Discrete Optimization 21 (2016) 25–41

Contents lists available at ScienceDirect

Discrete Optimization

www.elsevier.com/locate/disopt

Polyhedral results and a branch-and-cut algorithm for the double traveling Salesman problem with multiple stacks

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

Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS (UMR 7030), F-93430, Villetaneuse, France

ARTICLE INFO

Article history:
Received 1 August 2015
Received in revised form 25 April 2016
Accepted 26 April 2016
Available online 5 June 2016

MSC:
90B06
90C10
90C27
90C57

Keywords:
Traveling Salesman problem
Multiple stacks
Polyhedral study
Branch-and-cut

ABSTRACT

In the double TSP with multiple stacks, one performs a Hamiltonian circuit to pick up n items, storing them in a vehicle with s stacks of finite capacity q satisfying last-in-first-out constraints, and then delivers every item by performing a Hamiltonian circuit. We introduce an integer linear programming formulation with arc and precedence variables. We show that the underlying polytope shares some polyhedral properties with the ATSP polytope, which let us characterize large number of facets of our polytope. We convert these theoretical results into a branch-and-cut algorithm for the double TSP with two stacks. Our algorithm outperforms the existing exact methods and solves instances that were previously unsolved.

In this paper, we study a generalization of the Traveling Salesman Problem (*TSP*), namely the *double TSP with multiple stacks*. 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 will assume that the pickup and the delivery cities are very far from each other, thus the pickup phase has to be entirely completed before the delivery phase starts. The pickup (resp. delivery) phase consists in performing a Hamiltonian circuit, *i.e.*, starting from a depot, the n pickup (resp. delivery) locations have to be visited in sequence 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 according to its capacity and no rearrangement of the stacks is allowed. During the delivery circuit the stacks are unloaded following a last-in-first-out policy, that is, only items currently on the top of their stack can be delivered. The goal is to find the pickup and delivery circuits which minimize the total traveled distance, subject to the last-in-first-out consistency.

* Corresponding author. Tel.: +33 1 49 40 4071; fax: +33 1 48 26 0712.
E-mail address: roberto.wolfler@lipn.fr (R. Wolfler Calvo).

http://dx.doi.org/10.1016/j.disopt.2016.04.005
1572-5286/© 2016 Elsevier B.V. All rights reserved.

The double TSP with multiple stacks is introduced in [1] as a fleet management project initiated in cooperation with a software company. The problem arises from real-world applications. As the authors point out in [1], the items to be transported are usually standardized Euro Pallets, which are identical from a packing point of view. Moreover, repacking is not allowed because of insurance issues.

The double TSP with multiple stacks is NP-hard since, when the vehicle has only one stack, it corresponds to the Asymmetric Traveling Salesman Problem (*ATSP*): indeed, in this case, due to the last-in-first-out policy, the delivery circuit is nothing but the pickup circuit performed in the reverse order. Moreover, deciding if a given pair of pickup and delivery circuits satisfies the last-in-first-out policy is NP-complete [2]. It becomes polynomial when the number of stacks is fixed [3] or if the stacks have infinite capacity [4,2].

Since its first appearance, the double TSP with multiple stacks has received increasing attention. Both exact algorithms and heuristics have been designed for this problem over the past few years. Regarding the exact algorithms, in [5,6], the authors design a procedure to iteratively generate the k-best ATSP pickup and delivery solutions and to find the best combination satisfying the last-in-first-out consistency. Several exponential and polynomial size mixed integer linear programming formulations have been proposed and tested in branch-and-cut frameworks [7,8]. An additive branch-and-bound algorithm [9] has been developed for the case with two stacks. In [10], the authors adapt a branch-and-cut algorithm for the pickup and delivery TSP with multiple stacks to the double TSP with multiple stacks.

From a computational point of view, these algorithms clearly show that the double TSP with multiple stacks is extremely hard to solve with exact methods. In particular, the difficulty of the problem increases with the capacity of the stacks [8]. As a consequence, given a number of items equal to the total capacity, the hardest case is the double TSP with two stacks. Currently, no algorithm efficiently solves instances with capacity greater than seven.

In this paper, we first focus on the double TSP with multiple stacks of infinite capacity. Section 1 is devoted to notation and definitions. In Section 2, we introduce an integer linear programming formulation with arc and precedence variables. We then show in Section 3 that the underlying polytope shares some polyhedral properties with the ATSP polytope. These links let us characterize a super-polynomial number of facets of our polytope. Afterwards, in Section 4, we strengthen our formulation by exploiting the last-in-first-out consistency of the pickup and delivery circuits. In Section 5, we convert these theoretical results into a branch-and-cut algorithm for the double TSP with two stacks. It turns out that our algorithm outperforms the existing exact methods and solves new instances of the benchmark from the literature—see Section 6.

1. Definitions

Given a set $S \subseteq \mathbb{R}^m$, $\mathrm{conv}(S)$ is the convex hull of S; the symbol $\dim(S)$ denotes the dimension of the affine hull of S. Given $S \subseteq \mathbb{R}^n \times \mathbb{R}^d$, its projection into $\mathbb{R}^n$ is the set $proj_x(S) = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^d$ such that $(x, y) \in S\}$. The projection $proj_y(S)$ of S into $\mathbb{R}^d$ is defined in an analogous manner. For S a finite set, $x \in \mathbb{R}^{|S|}$ and $H \subseteq S$, we write $x(H)$ for $\sum_{h \in H} x_h$.

We denote by G_n the complete digraph having $V = \{0, \ldots, n\}$ as vertex set and $A = \{(i, j) : i \neq j \in V\}$ as arc set. A *circuit* of G_n is a set of arcs that induces a connected subgraph in which every vertex has exactly one entering and one leaving arc. Its *length* is the number of arcs it contains. A circuit is said *Hamiltonian* if its corresponding subgraph contains all the vertices. The *reverse* of a circuit C, denoted by $\overleftarrow{C}$, is the circuit composed of the opposite arcs of C.

A relation $\prec$ on $\{1, \ldots, n\}$ is a *linear ordering* if it is reflexive, antisymmetric, transitive and total. Such a relation is represented by an order $v_1, \ldots, v_n$ of $\{1, \ldots, n\}$, where $v_i \prec v_j$ whenever $i < j$. It is noteworthy that a Hamiltonian circuit $C = \{(0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n), (v_n, 0)\}$ of G_n induces a linear ordering $v_1, \ldots, v_n$ of $\{1, \ldots, n\}$. This ordering will be denoted by $\prec_C$. Moreover, the converse holds because G_n is complete. Hence, such a Hamiltonian circuit C will also be written $C = 0, v_1, \ldots, v_n, 0$.

Given two integers s and q, two Hamiltonian circuits are *s, q-consistent* if there exists an *s, q-loading plan* for them, that is, there exists a partition of $\{1, \ldots, n\}$ into s classes of size at most q such that each class can be ordered as a subsequence of both linear orderings induced by one circuit and the reverse of the other one. A couple of Hamiltonian circuits will be said s, q-consistent if the pair is. When q is infinite, we write *s-consistent.*

Given a Hamiltonian circuit C of G_n, we call *characteristic point* of C the point $(\chi^C, \gamma^C) \in \{0,1\}^{n(n+1)} \times \{0,1\}^{n(n-1)}$ defined by:

$$\chi^C_{ij} = \begin{cases} 1 & \text{if } (i,j) \in C, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for all } i \neq j \in V,$$

$$\gamma^C_{ij} = \begin{cases} 1 & \text{if } i \prec_C j, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for all } i \neq j \in V \setminus \{0\}.$$

When no confusion may arise, we will use the same terminology for the different representations described above; for instance, we will say that a family of Hamiltonian circuits is affinely independent whenever their characteristic points are.

We now describe the double TSP with multiple stacks in terms of graphs. An instance of this problem with n items is defined on the digraph G_n by two cost vectors c^P and c^D on its arcs, and a number s of identical stacks of capacity q. The digraph G_n models both cities; vertex 0 is the depot and the other ones are the locations where the items have to be picked up or delivered. The vectors c^P and c^D represent the distances between the locations of the pickup and delivery cities, respectively. The pickup and the delivery circuits are two Hamiltonian circuits of G_n. In the double TSP with multiple stacks, one seeks a solution of minimum cost, that is, a pair of s, q-consistent Hamiltonian circuits C_1 and C_2 whose cost $c^P(C_1) + c^D(C_2)$ is minimum.

2. A new integer linear programming formulation

In this section, we introduce an integer linear programming formulation with arc and precedence variables for the double TSP with multiple stacks of infinite capacity.

Given costs on the arcs of G_n, the well-known ATSP consists in finding a Hamiltonian circuit of G_n with minimum total cost. The ATSP has been widely studied over the past few decades and many mathematical formulations have been proposed—see *e.g.*, [11] for a recent survey. A point (x, y) is the characteristic point of some Hamiltonian circuit if and only if it satisfies the following system of constraints—see [12]:

$$\sum_{j=0}^{n} x_{ij} = 1 \qquad \text{for all } i \in V, \tag{1}$$

$$\sum_{i=0}^{n} x_{ij} = 1 \qquad \text{for all } j \in V, \tag{2}$$

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

$$y_{ij} + y_{jk} + y_{ki} - x_{ji} \geq 1 \qquad \text{for all } i \neq j \neq k \neq i \in V \setminus \{0\}, \tag{4}$$

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

$$x_{0i} \leq y_{ij} \qquad \text{for all } i \neq j \in V \setminus \{0\}, \tag{6}$$

$$x_{i0} \leq y_{ji} \qquad \text{for all } i \neq j \in V \setminus \{0\}, \tag{7}$$

$$y_{ij} \in \{0,1\} \qquad \text{for all } i \neq j \in V \setminus \{0\}, \tag{8}$$

$$x_{ij} \in \{0,1\} \qquad \text{for all } i \neq j \in V. \tag{9}$$

Throughout this paper, variables x are called *arc variables*, and variables y *precedence variables*. The convex hull of the points satisfying (1)–(9) is called *Precedence ATSP polytope – PATSP polytope* for short – and is denoted by $PATSP_n$. Under the integrality constraints (8) and (9), constraints (1) and (2) impose that each vertex of a Hamiltonian circuit C has exactly one entering and one leaving arc. Constraints (3) and (4) respectively describe the antisymmetry and the transitivity of the linear ordering $\prec_C$ and (5) implies that if $(i,j) \in C$ then $i \prec_C j$. Inequalities (6) (resp. (7)) indicate that if arc $(0,i)$ (resp. $(i,0)$) belongs to C, then i is the first (resp. last) vertex of the linear ordering C induces. We add (6) and (7) for later convenience; note that they are not necessary to this formulation.

The correspondence between Hamiltonian circuits in a complete digraph and linear orderings implies the following.

Remark 2.1. For $n \geq 2$, $proj_y\{(x,y) :$ (1)–(9) are satisfied$\}$ is the set of incidence vectors of the linear orderings of $\{1,\ldots,n\}$. Its convex hull is the *Linear Ordering Polytope*, see *e.g.*, [13].

The s-consistency of a pair of Hamiltonian circuits is characterized by Proposition 2.2, which independently appeared in [4,2]—see also [14] for a short proof.

Proposition 2.2 (*[4,2]*)**.** *Two Hamiltonian circuits of G_n are s-consistent if and only if less than $s+1$ vertices of $V \setminus \{0\}$ appear in the same order in both circuits.*

To describe the solutions to the double TSP with multiple stacks, we introduce the set of variables (x^P, y^P, x^D, y^D), where (x^P, y^P) and (x^D, y^D) are the variables used to describe the pickup and delivery Hamiltonian circuit, respectively.

The s-consistency is enforced by constraints (10) below:

$$\sum_{i=1}^{s}(y^P_{j_i j_{i+1}} + y^D_{j_i j_{i+1}}) \geq 1 \quad \text{for all distinct } j_1,\ldots,j_{s+1} \in \{1,\ldots,n\}. \tag{10}$$

Indeed, by Proposition 2.2, if two Hamiltonian circuits C_1 and C_2 are not s-consistent, then there exists a sequence $j_1,\ldots,j_{s+1}$ which is a subsequence of both C_1 and C_2. Remark 2.1 implies $\gamma^{C_1}_{j_i j_{i+1}} = \gamma^{C_2}_{j_i j_{i+1}} = 1$, for $i = 1,\ldots,s$. By constraints (3), this implies $\sum_{i=1}^{s}(\gamma^{C_1}_{j_{i+1} j_i} + \gamma^{C_2}_{j_{i+1} j_i}) = 0$ which contradicts constraints (10). We call constraints (10) *s-consistency constraints*.

The integer linear programming formulation we propose to model the double TSP with multiple stacks of infinite capacity is given by constraints (1)–(9), each expressed in the sets of variables (x^P, y^P) and (x^D, y^D), along with constraints (10). The convex hull of these solutions is referred to as the *DTSPMS polytope*, and denoted by $DTSPMS_{n,s}$. Its vertices are the *characteristic points* of the s-consistent couples of Hamiltonian circuits (C_1, C_2), denoted by $(\chi^{C_1}, \gamma^{C_1}, \chi^{C_2}, \gamma^{C_2})$.

It is worth noticing that our formulation ensures the existence of an s-loading plan for each solution. If required, one can explicitly construct such an s-loading plan in polynomial time with respect to the number of items n (in particular, independently from the number of stacks), as shown in [4,2].

3. Polyhedral links with the PATSP polytope

The main result of this section states that the DTSPMS polytope inherits all the facets of the PATSP polytope, see Theorem 3.2. This is a desirable property since the latter has $2^{\Omega(\sqrt{n})}$ facets [15]. Hence, our result characterizes a super-polynomial number of facets of the DTSPMS polytope. To prove this theorem, we will use a number of intermediate results which are postponed to the Appendix, due to their technicality. We only mention here Proposition 3.1 which provides the dimension of the PATSP polytope—see the Appendix for the proof. To the best of our knowledge, this is a new result.

Proposition 3.1. *If $n \geq 5$, then* $\dim(PATSP_n) = \frac{3n^2-3n-2}{2}$.

We now sketch the proof of Theorem 3.2. This result asserts that every facet-defining inequality of the PATSP polytope induces two facet-defining inequalities for the DTSPMS polytope. The detailed proof can be found in the Appendix.

Theorem 3.2. *For $n \geq 5$ and $s \geq 2$, if $ax + by \geq c$ defines a facet of $PATSP_n$, then $ax^T + by^T \geq c$ defines a facet of $DTSPMS_{n,s}$, for $T = P, D$.*

Sketch of proof. We denote by d_n the dimension of the PATSP polytope. Let $ax + by \geq c$ be a facet-defining inequality for the PATSP polytope. We now prove, without loss of generality, that $\mathcal{F} = \{(x^P, y^P, x^D, y^D) \in DTSPMS_{n,s} \colon ax^P + by^P = c\}$ is a facet of the DTSPMS polytope.

First, since $ax + by \geq c$ is a valid inequality for $PATSP_n$ and

$$DTSPMS_{n,s} \subseteq PATSP_n \times PATSP_n \quad \text{for all } s \geq 2, \tag{11}$$

$\mathcal{F}$ is a face of $DTSPMS_{n,s}$. By definition, there exist two Hamiltonian circuits C_1 and C_2 of the PATSP polytope such that exactly one of them satisfies $ax + by \geq c$ with equality. Since $(C_i, \overleftarrow{C_i})$ is a point of the DTSPMS polytope for $i = 1, 2$, the face $\mathcal{F}$ is a proper face of the DTSPMS polytope. Hence, $\dim(\mathcal{F}) \leq \dim(DTSPMS_{n,s}) - 1$. By (11), we have $\dim(DTSPMS_{n,s}) \leq 2d_n$, implying that $\dim(\mathcal{F}) \leq 2d_n - 1$.

To end the proof, we provide a family $\mathcal{S}$ of $2d_n$ affinely independent points of $\mathcal{F}$. Since $DTSPMS_{n,2} \subseteq DTSPMS_{n,s}$ for $s \geq 2$, it is enough to exhibit these $2d_n$ affinely independent points for the case $s = 2$.

The inequality $ax+by \geq c$ being facet-defining for the PATSP polytope, there exist d_n affinely independent Hamiltonian circuits $P_1, \ldots, P_{d_n}$ satisfying this inequality with equality. By relabeling the vertices, we may assume that P_1 is the circuit $0, n, n-1 \ldots, 2, 1, 0$.

To start the construction of the family $\mathcal{S}$, remark that the face of the DTSPMS polytope $\mathcal{F}' = \{(x^P, y^P, x^D, y^D) \in DTSPMS_{n,s} \colon (x^P, y^P) = (\chi^{P_1}, \gamma^{P_1})\}$ has dimension $d_n - 3$. Indeed, $\dim(\mathcal{F}') \leq d_n$ by definition. Since the last two vertices of P_1 are 2 and 1, if a Hamiltonian circuit C forms a 2-consistent pair with P_1 and $2 \prec_C 1$, then by Proposition 2.2, vertex 2 is the first vertex of C. This implies that $x^D_{02} \geq y^D_{21}$ is valid for $\mathcal{F}'$. Inequality (6) associated with $i = 2$ and $j = 1$, gives that equality actually holds. Hence

$$x^D_{02} = y^D_{21}, \tag{12}$$

is valid for $\mathcal{F}'$. The validity of (13) and (14) below can be proved similarly.

$$x^D_{(n-1)0} = y^D_{n(n-1)}, \tag{13}$$

$$x^D_{n1} = y^D_{n1}. \tag{14}$$

Equalities (12)–(14) are independent from the ones of $PATSP_n$. Therefore, $\dim(\mathcal{F}') \leq d_n - 3$, and the equality holds by Lemma A.1—see the Appendix.

Let $\mathcal{S} = \{(P_1, D_1), \ldots, (P_1, D_{d_n-2})\}$ be a family of affinely independent points of $\mathcal{F}'$. The 2-consistent couples $(P_i, \tilde{D}_i)$, $i = 2, \ldots, d_n$, where $\tilde{D}_i$ is obtained by perturbing $\overleftarrow{P_i}$ so that $(P_i, \tilde{D}_i)$ satisfies (12)–(14), are added to $\mathcal{S}$. This addition preserves the affine independence of $\mathcal{S}$, because the Hamiltonian circuits P_j, for $j = 1, \ldots, d_n$, are affinely independent. Since $\mathcal{F}' \subseteq \mathcal{F}$, we have $\mathcal{S} \subseteq \mathcal{F}$. Moreover, the $2d_n - 3$ points of $\mathcal{S}$ all satisfy (12)–(14) with equality.

To conclude, we need to find three more points of $\mathcal{F}$ that, together with those of $\mathcal{S}$, form an affinely independent family. This is done by exhibiting three points, each of them violating a different equality among (12)–(14). □

A result of this flavor can be found in [14]. More precisely, it is proved that every facet of $proj_x(PATSP_n)$ induces two facets of $proj_{(x^P,x^D)}(DTSPMS_{n,s})$. Nevertheless, their result cannot be applied to derive facet-defining inequalities involving precedence variables for our formulation.

We want to stress that several families of inequalities that are valid for the PATSP polytope have previously been introduced—see *e.g.*, [16,17]. To the best of our knowledge, it is still unknown whether those inequalities are facet-defining.

4. Strengthening the formulation

Theorem 3.2 yields a super-polynomial number of facets of the DTSPMS polytope yet they are not sufficient to characterize the convex hull. Here we provide a new family of inequalities in order to strengthen the linear relaxation of our formulation.

Consider the following example of the double TSP with multiple stacks. Suppose that there are 5 items and 2 stacks. Let c^P and c^D be such that $P^\star = 0, 1, 2, 3, 4, 5, 0$ and $D^\star = 0, 1, 2, 5, 4, 3, 0$ are Hamiltonian circuits which minimize c^P and c^D, respectively. Suppose also that costs are *symmetric*, that is, $c^P_{ij} = c^P_{ji}$ and $c^D_{ij} = c^D_{ji}$ for all $i \neq j \in V$. Neither $(P^\star, D^\star)$ nor $(P^\star, \overleftarrow{D^\star})$ is 2-consistent, and the point $S^\star = (P^\star, \frac{1}{2}D^\star + \frac{1}{2}\overleftarrow{D^\star})$ does not belong to the DTSPMS polytope. However, $S^\star$ belongs to the linear relaxation of our formulation and is actually one of its extreme points; the point $S^\star$ is also an optimal solution to this linear relaxation with respect to c^P and c^D. Moreover, no facet given by Theorem 3.2 can cut off $S^\star$ since both $P^\star$ and $\frac{1}{2}D^\star + \frac{1}{2}\overleftarrow{D^\star}$ belong to the PATSP polytope.

More generally, for symmetric costs, the value of the linear relaxation of our formulation cannot be better than the value obtained by independently optimizing the pickup and delivery circuits, that is, by independently solving two symmetric Traveling Salesman problems. The following family of valid inequalities strengthens the linear relaxation of our formulation by cutting off such extreme points.

Proposition 4.1. *Let C be a circuit of $G_n \setminus \{0\}$ with $|C| \geq s + 1$. Then the inequality*

$$y^P(C) + y^D(C) \geq \left\lceil \frac{|C|}{s} \right\rceil \tag{15}$$

is valid for $DTSPMS_{n,s}$.

Proof. For each arc a of C, consider the inequality (10) associated with the s consecutive arcs of C, starting from a. Summing these $|C|$ inequalities and dividing by s yields

$$y^P(C) + y^D(C) \geq \frac{|C|}{s}. \tag{16}$$

The vertices of $DTSPMS_{n,s}$ being integer, we round up the right hand side of (16) to get (15). □

Inequality (15) is a *circuit inequality of order* $|C|$. When $s = 2$ and $|C|$ is odd, we call it an *odd circuit inequality.* Such inequality is redundant when the length of the circuit is a multiple of s. When $s = 2$ and $|C| = 3$, it is dominated by the sum of the transitivity constraints (4) associated with C.

The extreme point $S^\star$ of the example given in the beginning of this section is cut off by an odd circuit inequality. Indeed, consider the circuit $H = 5, 4, 3, 2, 1, 5$. We have $\gamma^{P^\star}(H) = 0$ and $\frac{1}{2}\gamma^{D^\star}(H) + \frac{1}{2}\gamma^{\overleftarrow{D^\star}}(H) = \frac{5}{2}$, thus less than the value 3 required by the odd circuit inequality associated with H.

The validity of inequalities (15) can also be proven using the results of Sassano on the set covering polytope [18]. The *set covering polytope* associated with a 0/1 matrix A is the convex hull of the 0/1 solutions to $Ax \geq \mathbf{1}$. Let S be the set of points $(y^P, y^D) \in \{0,1\}^{n(n-1)} \times \{0,1\}^{n(n-1)}$ satisfying inequalities

(10) together with

$$y^T_{ij} + y^T_{ji} \geq 1 \quad \text{for all } i \neq j \in V \setminus \{0\} \text{ and } T = P, D, \tag{17}$$

$$y^T_{ij} + y^T_{jk} + y^T_{ki} \geq 1 \quad \text{for all } i \neq j \neq k \neq i \in V \setminus \{0\}. \tag{18}$$

Clearly, conv(S) is a set covering polytope, and $proj_{(y^P,y^D)}(DTSPMS_{n,s})$ is one of its faces by Remark 2.1 and because (17) is a relaxation of (3). Then, for each odd circuit C, the inequality $y^P(C) \geq \left\lceil \frac{|C|}{s} \right\rceil$ is the so-called *s-rose inequality of order* $|C|$ of the restriction of conv(S) to the variable set y^P. Using the Lifting Theorem 4.1 of [18], one can lift this inequality into the circuit inequality associated with C.

5. Branch-and-cut algorithm for the double TSP...

As explained in the introduction, given a number of items equal to the total capacity, the hardest case is the double TSP with two stacks. We develop a specific branch-and-cut algorithm for this case based on the formulation of Section 2. From the theoretical results of the previous sections, we first derive a branch-and-cut algorithm for the double TSP with two stacks of infinite capacity. We then modify this algorithm to take into account stacks of finite capacity. We refer the reader to [19] for a thorough description of branch-and-cut algorithms.

5.1. ...with two stacks of infinite capacity

Here we consider the double TSP with two stacks of infinite capacity. Specifying $s = 2$ in the formulation of Section 2 provides a formulation for this problem. We embed this formulation into a branch-and-cut algorithm.

To speed up our branch-and-cut algorithm, we add the 2-consistency inequalities and the transitivity inequalities in a dynamic way. Moreover, we strengthen our formulation with the odd circuit inequalities and two families of inequalities for the PATSP polytope: the subtour elimination constraints [20] and the 2-simple cut inequalities [16]. The *subtour elimination constraints* are the following:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad S \subset V \text{ such that } S \neq \emptyset. \tag{19}$$

The 2-*simple cut inequalities* include three (non-equivalent) families of inequalities, defined for each pair $i, j \in \{1, \dots, n\}$ as follows:

$$y_{ij} \leq \sum_{r \in V \setminus (S \cup \{j\})} \sum_{s \in S} x_{rs} \quad S \subseteq V \text{ such that } 0, j \notin S,\ i \in S \tag{20}$$

$$y_{ij} \leq \sum_{r \in V \setminus (S \cup \{0\})} \sum_{s \in S} x_{rs} \quad S \subseteq V \text{ such that } 0, i \notin S,\ j \in S \tag{21}$$

$$y_{ij} \leq \sum_{r \in V \setminus (S \cup \{i\})} \sum_{s \in S} x_{rs} \quad S \subseteq V \text{ such that } i, j \notin S,\ 0 \in S. \tag{22}$$

These constraints are valid due to the following observation. Given a Hamiltonian circuit C and two distinct vertices i and j of $V \setminus 0$, if i precedes j in C then there exists a path in C from 0 to i not containing j (constraint (20)), a path from i to j not containing 0 (constraint (21)) and a path from j to 0 not containing i (constraint (22)).

We mention that, even if there exist several other families of inequalities that are valid for the PATSP polytope, they do not seem to be efficient from a computational point of view [16].

When costs are symmetric, the points (P, D) and $(\overleftarrow{P}, \overleftarrow{D})$ have the same value. Since the instances we consider for tests have symmetric costs, we add the following equation to break this symmetry

$$y^P_{12} = 1. \tag{23}$$

We initiate our branch-and-cut algorithm with the starting formulation composed of inequalities (1)–(3), (5)–(7) and (23). The solutions to our problem are precisely the integer points satisfying the constraints from the starting formulation together with (4) and (10). At each node of the branch-and-cut tree, we perform a cutting plane phase by separating in that order the following families of inequalities:

- 2-consistency constraints (10),
- subtour elimination constraints (19),
- 2-simple cut inequalities (20)–(22),
- transitivity constraints (4),
- odd circuit inequalities (15).

The separation of each family is done when separating the previous ones yielded no violated constraint. We now describe the separation algorithms for each family with respect to the optimal solution $\bar{z} = (\bar{x}^1, \bar{y}^1, \bar{x}^2, \bar{y}^2)$ of the current linear relaxation.

2-*consistency constraints.* Since $s = 2$, there are $\Theta(n^3)$ inequalities of type (10). Consider $\bar{\jmath} \in V \setminus \{0\}$. Let $\bar{\imath}$ and $\bar{k}$ be the vertices minimizing $\bar{y}^1_{i\bar{\jmath}} + \bar{y}^2_{i\bar{\jmath}}$ and $\bar{y}^1_{\bar{\jmath}k} + \bar{y}^2_{\bar{\jmath}k}$, respectively. If $\bar{y}^1_{\bar{\imath}\bar{\jmath}} + \bar{y}^2_{\bar{\imath}\bar{\jmath}} + \bar{y}^1_{\bar{\jmath}\bar{k}} + \bar{y}^2_{\bar{\jmath}\bar{k}} < 1$, then the inequality (10) associated with $\bar{\imath}$, $\bar{\jmath}$ and $\bar{k}$ is violated because (3) guarantees that $\bar{\imath} \neq \bar{k}$. Otherwise, no inequality (10) associated with i, $\bar{\jmath}$ and k is violated by $\bar{z}$. Applying this procedure to each $\bar{\jmath} \in V \setminus \{0\}$ gives an exact separation of the 2-consistency constraints in $O(n^2)$.

Subtour elimination constraints. We use the Hao–Orlin algorithm [21] which provides a minimum cut in $O(n^3)$.

2-*simple cut inequalities.* We use the $O(n^5)$ exact separation algorithm of [16].

Transitivity constraints. The separation is done in $O(n^3)$ by enumeration.

Odd circuit inequalities. By the order of our separation procedure, the point $\bar{z}$ satisfies (10). Given an odd circuit C of $G_n \setminus \{0\}$, we have $\bar{y}^1(C) + \bar{y}^2(C) \geq \left\lceil \frac{|C|}{2} \right\rceil$ if and only if $2(\bar{y}^1(C) + \bar{y}^2(C)) \geq |C| + 1$. The latter can be rewritten as $w(C) \geq 1$, where $w_{ij} = 2(\bar{y}^1_{ij} + \bar{y}^2_{ij}) - 1$, for all $i \neq j \in V \setminus \{0\}$. Now, one of inequalities (15) is violated if and only if there exists an odd circuit of the weighted graph $(G_n \setminus \{0\}, w)$ of weight less than 1. Since $\bar{z}$ satisfies (10), it also satisfies (16). Therefore, $(G_n \setminus \{0\}, w)$ contains no circuit of negative weight. Finding a minimum odd circuit in such a weighted graph can be done in $O(n^4)$ [22], hence so can the separation of odd circuit inequalities.

5.2. *. . . with two stacks of finite capacity*

When the stacks have a finite capacity q, inequalities (10) are no more sufficient to ensure the $2, q$-consistency of couples of Hamiltonian circuits. In this case, we obtain a formulation by adding a family of constraints from [7] to the formulation of Section 2, namely the *tournament constraints*.

We briefly describe these constraints. Given a path R of G_n, denote by $T(R)$ the set of arcs (i, j) such that R visits i before j. Let R_P and R_D be two paths of G_n such that if $\tilde{P}$ and $\tilde{D}$ are Hamiltonian circuits with $\tilde{P} \supseteq R_P$ and $\tilde{D} \supseteq R_D$, then $(\tilde{P}, \tilde{D})$ is not $2, q$-consistent. Then, the inequality

$$x^P(T(R_P)) + x^D(T(R_D)) \leq |R_P| + |R_D| - 1, \tag{24}$$

is satisfied by the characteristic points of $2, q$-consistent pairs of Hamiltonian circuits.

We modify the branch-and-cut algorithm of Section 5.1 by separating constraints (24) whenever the current optimal solution is integer. For this separation, we use the exact separation algorithm of [7].

6. Computational results

All the exact algorithms for the double TSP with two stacks are tested on a benchmark of instances introduced in [1]. This benchmark contains the following instances. For every even n from 8 to 32, 20 instances are created with n items. Each instance is generated as follows. For both the pickup and delivery cities, n points are randomly generated in a square, the depot being in the center of the square. The costs are the Euclidean distances between these points, rounded to the nearest integers, and the objective is to minimize the total cost. The capacity of both stacks equals $\frac{n}{2}$.

We briefly review the different performances on these instances of the exact algorithms available in the literature for the double TSP with two stacks of capacity $\frac{n}{2}$. All the instances of the benchmark having at most 14 items are solved (to optimality), the best results being obtained by the algorithm of [7] using CPLEX 12 on a 3 GHz Intel Core Duo processor. Only three instances with 16 items are solved, by the algorithm of [10] using CPLEX 12 on a 2.2 GHz AMD Opteron 275 processor. All these algorithms fail to solve any other instance within a time limit of three hours.

Our branch-and-cut algorithm is implemented in C++ using CPLEX 12.5 [23] and is run on a 3.40 GHz Intel Core i7 processor in sequential mode, under Linux operating system. We use the strong branching rule of CPLEX and let CPLEX add its own generic families of cuts. All the graph-based routines used in the separation phase are implemented using the COIN-OR library LEMON [24]. An upper bound is initially given to our algorithm. This upper bound is obtained using the heuristic of [25]. We set a running time limit of three hours for each instance. Our experimental results on the benchmark described above are summarized in Table 1. This table is composed of two parts: the left one deals with infinite capacity; the right one takes into account a capacity of value $\frac{n}{2}$. These results come from the application of the branch-and-cut algorithms of Sections 5.1 and 5.2, respectively.

Each instance corresponds to a row. An instance is described by its name and its number of items, which appear in the corresponding column of the table. Both left and right parts of Table 1 are composed of four columns. Columns UB and LB respectively contain the value of the best feasible integer solution and the lower bound output by our algorithm within the time limit. Column CPU contains the CPU time spent expressed in seconds. Column Nodes contains the number of nodes of the branch-and-cut tree. Column Gap % contains the gap in percentage between the upper and lower bounds, which is 100(UB − LB)/UB. For each number of items, we add a row *Average* containing the average values of each column.

For the double TSP with two stacks of infinite capacity, the left part of Table 1 shows that our algorithm solves all the instances with up to 16 items and 8 out of the 20 instances with 18 items within the time limit of 3 h. The gap for the unsolved instances is at most 6.80%. The right part of Table 1 reports the computational results for the case of two stacks of capacity $\frac{n}{2}$. Our algorithm solves all the instances having up to 16 items and 7 out of the 20 instances with 18 items. The gap is less than 6.90% for the instances with 18 items which are not solved. For these instances, note that the number of nodes of the branch-and-cut tree does not exceed 56 000. We do not report the results for the instances with 20 items since both algorithms solve none of them.

As the right part of Table 1 shows, our algorithm outputs an optimal solution for all the instances already solved in the literature. In average, our algorithm is four times faster than the fastest available one which was run on a 3 GHz Intel Core Duo processor with Cplex 12 [7]. Moreover, we solve 24 previously unsolved instances: 17 instances with 16 items and 7 instances with 18 items.

The similarity of the left and right parts of Table 1 suggests that the finiteness of the stack capacity does not deeply modify the problem. Indeed, for the 67 instances which are solved by both algorithms, 37 have the

Table 1
Results of our branch-and-cut algorithms.

Instance	Items	Infinite capacity					Capacity $\frac{n}{2}$				
		UB	LB	CPU	Nodes	Gap %	UB	LB	CPU	Nodes	Gap %
R00	12	716	716.00	9.24	165	0.00	726	726.00	14.26	296	0.00
R01	12	741	741.00	6.79	175	0.00	741	741.00	6.81	175	0.00
R02	12	651	651.00	19.51	417	0.00	660	660.00	34.43	772	0.00
R03	12	690	690.00	1.80	15	0.00	690	690.00	1.81	15	0.00
R04	12	659	659.00	19.59	451	0.00	659	659.00	19.64	451	0.00
R05	12	627	627.00	35.78	724	0.00	631	631.00	41.73	858	0.00
R06	12	789	789.00	6.54	90	0.00	793	793.00	7.92	123	0.00
R07	12	589	589.00	7.02	138	0.00	593	593.00	8.05	171	0.00
R08	12	749	749.00	23.88	498	0.00	749	749.00	23.96	498	0.00
R09	12	686	686.00	4.53	84	0.00	692	692.00	4.91	107	0.00
R10	12	663	663.00	12.14	339	0.00	663	663.00	12.18	339	0.00
R11	12	622	622.00	12.27	244	0.00	625	625.00	13.90	286	0.00
R12	12	741	741.00	3.28	60	0.00	741	741.00	3.28	60	0.00
R13	12	683	683.00	2.93	49	0.00	694	694.00	5.88	116	0.00
R14	12	680	680.00	5.08	96	0.00	680	680.00	5.11	96	0.00
R15	12	624	624.00	14.26	315	0.00	628	628.00	21.71	493	0.00
R16	12	610	610.00	11.13	209	0.00	610	610.00	11.16	209	0.00
R17	12	780	780.00	44.91	971	0.00	780	780.00	45.09	971	0.00
R18	12	735	735.00	3.71	83	0.00	735	735.00	3.72	83	0.00
R19	12	782	782.00	20.40	490	0.00	789	789.00	36.14	874	0.00
Average				*13.24*	*280.65*	*0.00*			*16.08*	*349.65*	*0.00*
R00	14	766	766.00	118.39	1 544	0.00	774	774.00	168.63	2 299	0.00
R01	14	761	761.00	27.97	346	0.00	761	761.00	28.03	346	0.00
R02	14	690	690.00	129.33	1 648	0.00	690	690.00	129.62	1648	0.00
R03	14	791	791.00	52.13	593	0.00	791	791.00	52.25	593	0.00
R04	14	756	756.00	509.33	6 918	0.00	756	756.00	510.54	6 918	0.00
R05	14	773	773.00	127.46	1 589	0.00	775	775.00	130.30	1 650	0.00
R06	14	811	811.00	28.71	304	0.00	824	824.00	41.12	510	0.00
R07	14	693	693.00	28.21	319	0.00	697	697.00	31.97	378	0.00
R08	14	824	824.00	259.09	3 573	0.00	824	824.00	259.96	3 573	0.00
R09	14	733	733.00	5.93	58	0.00	739	739.00	9.96	126	0.00
R10	14	733	733.00	99.86	1 330	0.00	733	733.00	100.07	1 330	0.00
R11	14	719	719.00	238.89	2 975	0.00	725	725.00	266.40	3 443	0.00
R12	14	803	803.00	59.10	722	0.00	803	803.00	59.22	722	0.00
R13	14	743	743.00	36.56	508	0.00	746	746.00	41.03	590	0.00
R14	14	747	747.00	353.82	4 847	0.00	765	765.00	1329.61	16 658	0.00
R15	14	765	765.00	32.47	484	0.00	765	765.00	32.51	484	0.00
R16	14	685	685.00	31.57	376	0.00	685	685.00	31.63	376	0.00
R17	14	818	818.00	246.35	2 992	0.00	818	818.00	246.82	2 992	0.00
R18	14	774	774.00	94.40	1 325	0.00	774	774.00	94.57	1 325	0.00
R19	14	833	833.00	237.57	3 002	0.00	836	836.00	289.47	3 685	0.00
Average				*135.86*	*1772.65*	*0.00*			*192.69*	*2482.30*	*0.00*
R00	16	795	795.00	1498.13	12 002	0.00	804	804.00	2446.04	20 047	0.00
R01	16	794	794.00	169.58	1 467	0.00	794	794.00	169.61	1 467	0.00
R02	16	752	752.00	6688.66	51 700	0.00	752	752.00	7001.64	53 649	0.00
R03	16	855	855.00	1879.71	13 641	0.00	855	855.00	1883.62	13 680	0.00
R04	16	792	792.00	6616.13	52 883	0.00	801	801.00	10 499.83	84 698	0.00
R05	16	820	820.00	4248.95	32 078	0.00	823	823.00	4417.25	33 882	0.00
R06	16	900	900.00	988.01	8 057	0.00	906	906.00	1563.23	12 737	0.00
R07	16	756	756.00	130.26	958	0.00	756	756.00	130.44	958	0.00
R08	16	907	907.00	1526.68	12 634	0.00	909	909.00	1756.94	14 674	0.00
R09	16	796	796.00	99.46	789	0.00	800	800.00	149.97	1 197	0.00
R10	16	755	755.00	664.12	5 300	0.00	755	755.00	796.47	6 634	0.00
R11	16	759	759.00	909.18	7 377	0.00	777	777.00	3385.05	27 916	0.00
R12	16	825	825.00	653.00	5 264	0.00	825	825.00	653.90	5 264	0.00
R13	16	824	824.00	719.47	5 878	0.00	831	831.00	1160.47	9 537	0.00
R14	16	823	823.00	5892.60	41 223	0.00	823	823.00	5948.71	41 328	0.00
R15	16	807	807.00	568.39	4 549	0.00	807	807.00	566.59	4 549	0.00
R16	16	781	781.00	2347.62	18 234	0.00	781	781.00	2353.94	18 234	0.00
R17	16	852	852.00	2136.11	16 101	0.00	858	858.00	3338.61	25 070	0.00
R18	16	846	846.00	1289.01	10 532	0.00	846	846.00	1297.49	10 532	0.00
R19	16	882	882.00	1589.97	12 501	0.00	882	882.00	1910.27	15 243	0.00
Average				*2030.75*	*15 658.40*	*0.00*			*2571.50*	*20 064.80*	*0.00*

(*continued on next page*)

Table 1 (*continued*)

Instance	Items	Infinite capacity					Capacity $\frac{n}{2}$				
		UB	LB	CPU	Nodes	Gap %	UB	LB	CPU	Nodes	Gap %
R00	18	839	839.00	5128.95	28 232	0.00	839	839.00	5192.40	28 229	0.00
R01	18	825	825.00	1574.57	7 119	0.00	857	857.00	10 038.08	50 055	0.00
R02	18	793	750.06	10 800.00	46 046	5.42	793	750.20	10 800.00	46 183	5.40
R03	18	896	848.67	10 800.00	43 700	5.28	899	847.28	10 800.00	43 286	5.75
R04	18	832	781.50	10 800.00	44 790	6.07	832	781.42	10 800.00	44 570	6.08
R05	18	873	847.60	10 800.00	50 545	2.91	873	847.70	10 800.00	50 529	2.90
R06	18	930	930.00	9257.50	44 850	0.00	930	930.00	9394.32	45 417	0.00
R07	18	805	805.00	1488.97	7 918	0.00	805	805.00	1419.58	7 538	0.00
R08	18	962	907.68	10 800.00	43 758	5.65	962	907.75	10 800.00	43 456	5.64
R09	18	815	815.00	448.44	2 510	0.00	815	815.00	448.36	2 510	0.00
R10	18	856	825.04	10 800.00	44 155	3.62	856	825.12	10 800.00	44 355	3.61
R11	18	823	788.97	10 800.00	51 234	4.13	823	789.86	10 800.00	51 466	4.03
R12	18	871	871.00	4291.89	21 560	0.00	871	871.00	4734.43	23 903	0.00
R13	18	845	845.00	3455.85	19 047	0.00	860	852.53	10 800.00	55 775	0.87
R14	18	873	813.67	10 800.00	40 037	6.80	874	813.75	10 800.00	40 151	6.89
R15	18	869	834.64	10 800.00	47 370	3.95	869	834.89	10 800.00	47 597	3.93
R16	18	811	811.00	5499.46	28 197	0.00	819	819.00	7590.24	39 622	0.00
R17	18	900	840.50	10 800.00	38 099	6.61	900	840.57	10 800.00	38 076	6.60
R18	18	883	867.33	10 800.00	47 342	1.77	883	866.83	10 800.00	47 128	1.83
R19	18	909	893.13	10 800.00	51 974	1.75	909	894.68	10 800.00	52 628	1.58
Average				*8037.28*	*35 424.15*	*2.70*			*8960.87*	*40 123.70*	*2.75*

same optima. For the other ones, taking into account the capacity increases the cost of an optimal solution by 0.45% in average and by 3.88% in the worst case. From a computational point of view, all the instances which are solved with infinite capacity are also solved with finite capacity, except one. The exception is instance R13 with 18 items for which the branch-and-cut algorithm with finite capacity obtains a gap of 0.81%. Beside, when the optima are equal for both cases, the CPU time is almost identical. For all instances but R07 with 18 items, taking into account the capacity slightly increases the CPU time. For example, the average CPU time increases by less than 12% for the instances having 18 items.

We mention here that the heuristic of [25], used to get upper bounds for our algorithms, actually finds an optimal solution for all the instances with finite capacity we solve. However, running the same algorithms on the instances with 12, 14 and 16 items without using any upper bound does not affect drastically their performances. Indeed, all the instances but one are solved within the time limit. Moreover, the CPU time remains of the same order of magnitude for both algorithms.

These experimental results show that, with our approach, the finiteness of the capacity is not the major computational difficulty.

7. Concluding remarks

In this paper, we develop a branch-and-cut algorithm based on polyhedral results for the double TSP with multiple stacks. From a computational point of view, we solve many instances that were previously unsolved. Our approach, which mainly focuses on the consistency of Hamiltonian circuits when the stack capacity is infinite, is validated by the experimental results. Moreover, we think that our approach is promising to solve bigger instances, as the results show that the finiteness of the capacity is not the major computational difficulty and our branch-and-cut tree suffers from no combinatorial explosion. A possible direction to improve our algorithm is to strengthen our formulation by studying the facial structure of the PATSP polytope, thanks to Theorem 3.2.

Acknowledgments

We thank Martínez et al. [7], Côté et al. [25] and Casazza et al. [4] for kindly sharing their code with us.

Appendix

Proof of Proposition 3.1. The points of $PATSP_n$ satisfy the equalities (1)–(3). The rank of the matrix of equalities (1)–(3) being $\binom{n}{2} + 2n + 1 = \frac{n^2+3n+2}{2}$ (see *e.g.*, [26,13]), and the number of variables being $2n^2$, we get $d_n \leq \frac{3n^2-3n-2}{2}$. To prove the result, it is enough to find $\frac{3n^2-3n-2}{2} + 1$ affinely independent points of $PATSP_n$, for every n.

We proceed by induction. One can check by enumeration that the result holds for $n = 5$. Let us define for every $n \geq 5$, $k_n = \frac{3n^2-3n-2}{2}$ and, assuming that the proposition holds for $PATSP_n$, let us prove that it also does for $PATSP_{n+1}$.

By the inductive hypothesis, there exist $C_1 \dots C_{k_n+1}$ affinely independent Hamiltonian circuits of G_n. By inserting $n+1$ at the end of each C_i, we get a set of k_n+1 affinely independent Hamiltonian circuits of G_{n+1}. Since $k_{n+1} - k_n = 3n$, it suffices to complete this set with $3n$ new Hamiltonian circuits of G_n, maintaining the affine independence. The circuits are added in an iterative fashion. We indicate by $C_{(i,j)}$ a circuit that contains the arc (i, j) not belonging to any of the circuits added in the previous iterations; similarly, $C^\star_{(i,j)}$ indicates a circuit where i precedes j for the first time until the given iteration. Then, by construction, adding the $2n + 1$ circuits below, in the order they are presented, preserves the affine independence:

$$\begin{aligned}
&C_{(n,0)} = 0, 2, \dots, n-1, n+1, 1, n, 0\\
&C^\star_{(n+1,2)} = 0, 3 \dots, n-1, n+1, 1, 2, n, 0\\
&C^\star_{(n+1,i)} = 0, 2 \dots, i-1, i+1, \dots, n-1, n+1, 1, i, n, 0 \quad \text{for } i = 3, \dots, n-2\\
&C^\star_{(n+1,n-1)} = 0, 2 \dots, n-2, n+1, 1, n-1, n, 0\\
&C_{(n+1,i)} = 0, 1, \dots, i-1, n+1, i, \dots, n, 0 \quad \text{for } i = 2, \dots, n\\
&C_{(0,n+1)} = 0, n+1, 1, 2, \dots, n, 0\\
&C_{(2,0)} = 0, 1, 3, \dots, n-1, n+1, n, 2, 0\\
&\tilde{C}_1 = 0, 1, 3, \dots, n, n+1, 2, 0.
\end{aligned}$$

Adding $\tilde{C}_1$ maintains the affine independence since every previous circuit C such that $\gamma^C_{n(n+1)} = 1$ also verifies $\chi^C_{(n+1)0} = 1$. Hence, $\tilde{C}_1$ cannot be obtained as an affine combination of the previous circuits. Finally, we add the following $n-1$ circuits:

$$\begin{aligned}
&C_{(1,0)} = 0, 2, \dots, n+1, 1, 0\\
&C_{(i,0)} = 0, i+1, \dots, n+1, 1, \dots, i, 0 \quad \text{for } i = 3, \dots, n-1\\
&\tilde{C}_2 = 0, 2, \dots, n-1, n+1, n, 1, 0.
\end{aligned}$$

Adding $\tilde{C}_2$ preserves the affine independence since every previous circuit C such that $\gamma^C_{(n+1)1} = 1$ also verifies $\chi^C_{(n+1)1} = 1$. The whole family of circuits above forms an affinely independent set, and this concludes the proof. □

Lemma A.1. *The set of Hamiltonian circuits that are s-consistent with a fixed Hamiltonian circuit of G_n has dimension d_n for $n \geq 4$ and $s \geq 3$ and dimension $d_n - 3$ for $n \geq 5$ and $s = 2$.*

Proof. Let us fix a Hamiltonian circuit of G_n. We can assume that this circuit is $C^n = 0, n, \dots, 1, 0$, as otherwise we can relabel the vertices. Let us call $\mathcal{C}^{n,s}$ the set of Hamiltonian circuits that are s-consistent with C^n. We split the proof into two cases.

Case $s \geq 3$. The case $n = 4$ can be checked by enumeration, showing that $\dim(\mathcal{C}^{4,3}) = 17$. This implies $\dim(\mathcal{C}^{4,s}) = 17$ for every $s \geq 3$. Then we can apply induction, observing that the circuits constructed in the proof of Proposition 3.1 are s-consistent with C^{n+1} when $s \geq 3$.

Case $s = 2$. Note that in the case with two stacks and $n \geq 5$, the following three equalities are valid for the set $\mathcal{C}^{n,2}$ by Proposition 2.2:

$$x_{02} = y_{21} \tag{25}$$

$$x_{(n-1)0} = y_{n(n-1)} \tag{26}$$

$$x_{n1} = y_{n1}. \tag{27}$$

In addition, the Eqs. (25)–(27) are easily seen to be linearly independent. Therefore, adding (25)–(27) to Eqs. (1)–(3) we get $\dim(\mathcal{C}^{n,2}) \leq d_n - 3$.

Let $n \geq 5$ and let us prove by induction that $\dim(\mathcal{C}^{n,2}) = d_n - 3$. The base case $n = 5$ can be checked by enumeration. Assuming that the result holds for $\mathcal{C}^{n,2}$, let us prove it also for $\mathcal{C}^{n+1,2}$. By the inductive hypothesis, there exist $C_1 \dots C_{d_n-2}$ affinely independent Hamiltonian circuits of G_n that are 2-consistent with C^n. By inserting $n+1$ at the end of each C_i we get a set $\mathcal{C}$ of $d_n - 2$ independent Hamiltonian circuits of G_{n+1} that are 2-consistent with C^{n+1}. For later convenience, let us partition $\mathcal{C}$ in the two sets $\mathcal{C}_1 = \{C \in \mathcal{C} : \chi^C_{(n-1)(n+1)} = 1\}$ and $\mathcal{C}_2 = \mathcal{C} \setminus \mathcal{C}_1$.

Observe that if $C \in \mathcal{C}_2$, by (3) and (26) it follows that $\gamma^C_{(n-1)n} = 1$. As in the proof of Proposition 3.1, it suffices to complete $\mathcal{C}$ with $3n$ affinely independent circuits that are 2-consistent with C^{n+1}. For these new circuits, we use the notation from the proof of Proposition 3.1. By construction, the affine independence is ensured when the two circuits below are added in the following order:

$$C_{(n,0)} := 0, 2, \dots, n-1, n+1, 1, n, 0$$
$$C_{(n+1,n)} = 0, 1, \dots, n-1, n+1, n, 0.$$

Next, we add the circuit:

$$\tilde{C}_1 = 0, 2, \dots, n-1, 1, n+1, n, 0.$$

Claim A.2. *The set* $\mathcal{C} \cup \{C_{(n,0)}, C_{(n+1,n)}, \tilde{C}_1\}$ *is a set of affinely independent circuits.*

Proof. By contradiction, we may assume that $\tilde{C}_1$ is a combination of the circuits in $\mathcal{C} \cup \left\{C_{(n,0)}, C_{(n+1,n)}\right\}$. We indicate with λ^C the coefficient of the circuit C in such a combination. Since there is a direct arc from n to 0 only in $C_{(n,0)}$, $C_{(n+1,n)}$ and $\tilde{C}_1$, and a direct arc from $n+1$ to n only in $C_{(n+1,n)}$ and $\tilde{C}_1$, we get that $\lambda^{C_{(n,0)}} = 0$ and $\lambda^{C_{(n+1,n)}} = 1$. In addition $\chi^{\tilde{C}_1}_{(n-1)(n+1)} = 0$, then $\sum_{C \in \mathcal{C}_1} \lambda^C = -1$. Similarly, as $\chi^{\tilde{C}_1}_{(n+1)0} = 0$, we have also that $\sum_{C \in \mathcal{C}_1} \lambda^C + \sum_{C \in \mathcal{C}_2} \lambda^C = 0$, i.e., $\sum_{C \in \mathcal{C}_2} \lambda^C = 1$. But this would imply $\gamma^{\tilde{C}_1}_{(n-1)n} = 2$—a contradiction. □

Subsequently, we iteratively add the following $2n - 2$ circuits:

$$C^{\star}_{(n+1,2)} = 0, 3 \dots, n-1, n+1, 1, 2, n, 0$$
$$C^{\star}_{(n+1,i)} = 0, 2 \dots, i-1, i+1, \dots, n-1, n+1, 1, i, n, 0 \quad \text{for } i = 3, \dots, n-2$$
$$C^{\star}_{(n+1,n-1)} = 0, 2 \dots, n-2, n+1, 1, n-1, n, 0$$
$$C_{(n+1,i)} = 0, 1, \dots, i-1, n+1, i, i+1, \dots, n, 0 \quad \text{for } i = 2, \dots, n-1$$
$$C_{(0,n+1)} = 0, n+1, 1, 2, \dots, n, 0$$
$$C_{(2,0)} = 0, 1, 3, \dots, n, n+1, 2, 0.$$

By construction, the resulting set contains only affinely independent circuits. The next circuit we add is:

$$\tilde{C}_2 = 0, 3, \ldots, n, n+1, 1, 2, 0.$$

The circuit $\tilde{C}_2$ is independent of the previous ones because $\gamma_{n1}^{\tilde{C}_2} = 1$ and $\chi_{n1}^{\tilde{C}_2} = 0$, whereas each circuit C added in the previous steps such that $\gamma_{n1}^{C} = 1$ belongs to $\mathcal{C}$, and also verifies $\chi_{n1}^{C} = 1$, because of (27). Finally, the last $n-2$ circuits we add are the following ones:

$$\begin{aligned} C_{(1,0)} &= 0, 2, \ldots, n+1, 1, 0 \\ C_{(i,0)} &= 0, i+1, \ldots, n+1, 1, \ldots, i, 0 \quad \text{for } i = 3, 4, \ldots, n-1 \end{aligned}$$

whose independence again follows by construction. The $3n$ circuits above are 2-consistent with C^{n+1}, and this concludes the proof. □

Proposition A.3. *For $n \geq 5$ and $s \geq 2$, we have* $\dim(DTSPMS_{n,s}) = 2d_n$.

Proof. Given the inclusions $DTSPMS_{n,2} \subseteq DTSPMS_{n,s} \subseteq PATSP_n \times PATSP_n$, it is enough to prove the result for $s = 2$.

For $n \geq 5$, let $P_1 \ldots P_{d_n+1}$ be affinely independent Hamiltonian circuits of G_n. From Lemma A.1 there exist $D_1 \ldots D_{d_n-2}$ affinely independent Hamiltonian circuits such that $V_i = (P_1, D_i)$ are pairs of 2-consistent Hamiltonian circuits. By relabeling the vertices, we can assume that $P_1 = 0, n, \ldots, 1, 0$. Under this assumption, every D_i satisfies (25)–(27). Given P_j, for some $1 < j \leq d_n + 1$, we will now construct $\tilde{D}_j$ satisfying (25)–(27) such that $V_{d_n-3+j} = (P_j, \tilde{D}_j)$ is a pair of 2-consistent Hamiltonian circuits. From the affine independence of the P_j for $j = 1, \ldots, n$, it will follow that the characteristic points of the pairs $V_1 \ldots V_{2d_n-2}$, taken in the order given by the subscripts, will be affinely independent.

Roughly speaking, $\tilde{D}_j$ is obtained by perturbing $\overleftarrow{P_j}$ in such a way that Eqs. (25)–(27) are satisfied; we exploit the observation that $(P_j, \overleftarrow{P_j})$ is a pair of 2-consistent Hamiltonian circuits to find a perturbation such that also the new pair $(P_j, \tilde{D}_j)$ of circuits is 2-consistent.

In detail, if $2 \prec_{\overleftarrow{P_j}} n$, then $\tilde{D}_j$ is obtained from $\overleftarrow{P_j}$ by putting 2 at its beginning and n at its end. Note that $(P_j, \tilde{D}_j)$ is a solution to the double TSP with two stacks and $\tilde{D}_j$ verifies (25)–(27). Therefore, assume $n \prec_{\overleftarrow{P_j}} 2$. If also $1 \prec_{\overleftarrow{P_j}} 2$, then $\tilde{D}_j$ is obtained from $\overleftarrow{P_j}$ by moving n in its last position. Even in this case, $\tilde{D}_j$ is 2-consistent with P_j and verifies (25)–(27).

Now, let $n \prec_{\overleftarrow{P_j}} 2 \prec_{\overleftarrow{P_j}} 1$. Suppose $\overleftarrow{P_j} = 0, X_1, n, X_2, n-1, X_3, 2, X_4, 1, X_5, 0$ where the X_i's represent sequences of vertices. Then, a circuit 2-consistent with P_j and verifying (25)–(27), is $\tilde{D}_j = 0, X_1, X_2, n-1, n, 1, X_3, 2, X_4, X_5, 0$.

Lastly, if $n-1 \prec_{\overleftarrow{P_j}} n$ or $2 \prec_{\overleftarrow{P_j}} n-1$, let $\overleftarrow{P_j} = 0, X_1, X_2, 0$ where X_2 is the part of the circuit starting at node 2. In both cases, $\tilde{D}_j = 0, X_2, X_1, 0$ is 2-consistent with P_j and satisfies (25)–(27).

To conclude the proof, consider the following three Hamiltonian circuits[1]:

$$\begin{aligned} D_1^\star &= 0, 3, 2, 1, 4, \ldots, n-2, n, n-1, 0 \\ D_2^\star &= 0, 2, 1, n, n-1, 3, \ldots, n-2, 0 \\ D_3^\star &= 0, 2, n, 3, 1, 4, \ldots, n-1, 0. \end{aligned}$$

[1] In the case $n = 5$, we set $D_1^\star = 0, 3, 2, 1, 5, 4, 0$.

Note that the $D^\star_k$ above are well defined because $n \geq 5$. Setting $P^\star_k = \overleftarrow{D^\star}_k$, we get that the characteristic points of the pairs $V_{2d_n-2+k} = (P^\star_k, D^\star_k)$ are all independent with each other and with the previous points, because $D^\star_k$ only violates the kth equation of (25)–(27). The characteristic points of the pairs $V_1 \dots V_{2d_n+1}$ form a family of $2d_n + 1$ affinely independent points of $DTSPMS_{n,2}$. $\square$

Proof of Theorem 3.2. Let us consider an inequality $ax + by \geq c$ valid for the PATSP polytope, such that the set $\mathcal{F}' = \{(x, y) \in PATSP_n : ax + by = c\}$ is a facet of $PATSP_n$. Without loss of generality, let us show that $\mathcal{F} = \{(x^P, y^P, x^D, y^D) \in DTSPMS_{n,s} : ax^P + by^P = c\}$ is a facet of $DTSPMS_{n,s}$.

Note that $\mathcal{F}$ is a face of $DTSPMS_{n,s}$: indeed, $ax^P + by^P \geq c$ is valid for $DTSPMS_{n,s}$ because $DTSPMS_{n,s} \subseteq PATSP_n \times PATSP_n$ and $ax + by \geq c$ is valid for $PATSP_n$. To prove that $\mathcal{F}$ is a facet of $DTSPMS_{n,s}$ we will exhibit $2d_n$ affinely independent points in $\mathcal{F}$. Since $DTSPMS_{n,2} \subseteq DTSPMS_{n,s}$ for every $s \geq 2$, it is enough to prove the existence of these affinely independent points for $s = 2$.

By hypothesis there exist $P_1 \dots P_{d_n}$ affinely independent Hamiltonian circuits belonging to $\mathcal{F}'$. In addition, it is not restrictive to assume $P_1 = 0, n, \dots, 1, 0$. Under this assumption, and repeating the reasoning used in the proof of Proposition A.3, we can construct Hamiltonian circuits $D_1, \dots, D_{d_n-2}$ and $\tilde{D}_2, \dots, \tilde{D}_{d_n}$ such that the $2d_n - 3$ pairs of circuits

$$(P_1, D_1), \dots, (P_1, D_{d_n-2}), (P_2, \tilde{D}_2), \dots, (P_{d_n}, \tilde{D}_{d_n}) \tag{28}$$

are 2-consistent, and their characteristic points form an affine independent set of $\mathcal{F}$ and verify:

$$x^D_{02} = y^D_{21} \tag{29}$$

$$x^D_{(n-1)0} = y^D_{n(n-1)} \tag{30}$$

$$x^D_{n1} = y^D_{n1}. \tag{31}$$

Now, assume that $ax^P + by^P \geq c$ coincides with the inequality $x^P_{01} \geq 0$. In this case, the Theorem holds, since we can complete the set in (28) to an affine base of $\mathcal{F}$ by adding the pairs of Hamiltonian circuits $(P^\star_k, D^\star_k)$ as constructed in the proof of Proposition A.3. Indeed, their characteristic points satisfy $x^P_{01} = 0$ and the affine independence of the resulting set is seen as in Proposition A.3. The same also applies if $ax^P + by^P \geq c$ coincides with $x^P_{(n-1)0} \geq 0$ or with $x^P_{n0} \geq 0$.

Therefore, assume that $ax^P + by^P \geq c$ does not coincide with any of the inequalities $x^P_{01} \geq 0$, $x^P_{(n-1)0} \geq 0$, $x^P_{n0} \geq 0$.

By hypothesis, there exists a circuit $P^\star_1 \in \mathcal{F}'$ such that $\chi^{P^\star_1}_{01} = 1$. In other words, $P^\star_1 = 0, 1, X_1, 2, X_2, 0$ for some sequences of vertices X_1 and X_2. We first assume that $X_2 \neq \{n\}$. Then, if $X_2 \neq \emptyset$, we define $D^\star_1 = 0, \overleftarrow{X'}_2, 2, \overleftarrow{X'}_1, 1, n, 0$, where X'_1 and X'_2 are respectively obtained from X_1 and X_2 by removing n. If $X_2 = \emptyset$, then $P^\star_1 = 0, 1, X_3, n, X_4, 2, 0$ for some sequences X_3 and X_4. In this case, when $X_4 = \emptyset$ we define $D^\star_1 = 0, r, 2, \overleftarrow{X'}_3, n, 1, 0$, where X'_3 is obtained from X_3 by removing its last vertex r. Instead, if $X_4 \neq \emptyset$, we set $D^\star_1 = 0, t, 2, \overleftarrow{X'}_4, \overleftarrow{X}_3, n, 1, 0$, where X'_4 is obtained from X_4 by removing its first vertex t. In each case, $(P^\star_1, D^\star_1) \in \mathcal{F}$ (because it is a pair of 2-consistent Hamiltonian circuits) and its characteristic point violates (29) (while satisfying (30) and (31)). As a consequence, it cannot be in the affine subspace generated by the points in (28).

Similarly, there exists a Hamiltonian circuit $P^\star_2 \in \mathcal{F}'$ such that $\chi^{P^\star_2}_{(n-1)0} = 1$ (possibly $P^\star_2 = P^\star_1$). We distinguish the following two cases, where X_5, X_6, X_7 denote suitable sequences of vertices:

(1) $P^\star_2 = 0, X_5, n, X_6, 1, X_7, n-1, 0$

(2) $P_2^\star = 0, X_5, 1, X_6, n, X_7, n-1, 0$.

We define $D_2^\star = 0, 1, n, n-1, \overleftarrow{X}_7, \overleftarrow{X}_6, \overleftarrow{X}_5, 0$ in case (1), whereas in case (2), $D_2^\star = 0, n, 1, n-1, \overleftarrow{X}_7, \overleftarrow{X}_6, \overleftarrow{X}_5, 0$. In both cases, $(P_2^\star, D_2^\star) \in \mathcal{F}$ and the affine independence of its characteristic point with the previous ones is given by the fact that it violates only (30) while satisfying (29) and (31).

Moreover, let $P_3^\star \in \mathcal{F}'$ be a Hamiltonian circuit such that $\chi_{n0}^{P_3^\star} = 1$ ($P_3^\star$ may coincide with $P_1^\star$). Since $P_3^\star = 0, X_8, 1, X_9, n, 0$ for some sequence of vertices X_8 and X_9, defining $D_3^\star = 0, n, \overleftarrow{X}_9, \overleftarrow{X}_8, 1, 0$ gives that $(P_3^\star, D_3^\star) \in \mathcal{F}$. In addition, its characteristic point cannot be obtained as an affine combination of the previous ones, as it violates Eqs. (29)–(31).

Completing the set of circuits in (28) with the pairs $(P_i^\star, D_i^\star)$ yields a set of $2d_n$ affinely independent pairs of Hamiltonian circuits belonging to $\mathcal{F}$ and this concludes the proof for the case in which $X_2 \neq \{n\}$.

Finally, let us assume that $X_2 = \{n\}$ in $P_1^\star$, that is, $P_1^\star = 0, 1, X_1, 2, n, 0$. We define $D_1^\star = 0, 2, n-1, n, \overleftarrow{X}_1', 1, 0$ where X_1' is obtained from X_1 by removing $n-1$. Note that, since $n \geq 5$, X_1' is not empty. Now, $(P_1^\star, D_1^\star) \in \mathcal{F}$ and its characteristic vector violates Eq. (31), while satisfying Eq. (29) and Eq. (30). We construct the points $(P_2^\star, D_2^\star)$ and $(P_3^\star, D_3^\star)$ as done above. The same argument used for the case $X_2 \neq \{n\}$ shows that, also in the case $X_2 = \{n\}$, completing the set of circuits in Eq. (28) with the pairs $(P_i^\star, D_i^\star)$ yields a set of $2d_n$ affinely independent pairs of Hamiltonian circuits belonging to $\mathcal{F}$. This concludes the proof. □

References

[1] H.L. Petersen, O.B.G. Madsen, The double travelling salesman problem with multiple stacks—formulation and heuristic solution approaches, European J. Oper. Res. 198 (1) (2009) 139–147.

[2] S. Toulouse, R. Wolfler Calvo, On the complexity of the multiple stack TSP, kSTSP, in: Proceedings of the 6th Annual Conference on Theory and Applications of Models of Computation, TAMC'09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 360–369.

[3] F. Bonomo, S. Mattia, G. Oriolo, Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem, Theoret. Comput. Sci. 412 (45) (2011) 6261–6268.

[4] M. Casazza, A. Ceselli, M. Nunkesser, Efficient algorithms for the double traveling salesman problem with multiple stacks, Comput. Oper. Res. 39 (5) (2012) 1044–1053.

[5] R.M. Lusby, J. Larsen, Improved exact method for the double TSP with multiple stacks, Networks 58 (4) (2011) 290–300.

[6] R.M. Lusby, J. Larsen, M. Ehrgott, D. Ryan, An exact method for the double TSP with multiple stacks, Int. Trans. Oper. Res. 17 (5) (2010) 637–652.

[7] M.A.A. Martínez, J.-F. Cordeau, M. Dell'Amico, M. Iori, A branch-and-cut algorithm for the double traveling salesman problem with multiple stacks, INFORMS J. Comput. 25 (1) (2013) 41–55.

[8] H.L. Petersen, C. Archetti, M.G. Speranza, Exact solutions to the double travelling salesman problem with multiple stacks, Networks 56 (4) (2010) 229–243.

[9] F. Carrabs, R. Cerulli, M.G. Speranza, A branch-and-bound algorithm for the double travelling salesman problem with two stacks, Networks 61 (1) (2013) 58–75.

[10] J.-F. Côté, C. Archetti, M.G. Speranza, M. Gendreau, J.-Y. Potvin, A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks, Networks 60 (4) (2012) 212–226.

[11] R. Roberti, P. Toth, Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison, EURO J. Transp. Logist. 1 (1–2) (2012) 113–133.

[12] S.C. Sarin, H.D. Sherali, A. Bhootra, New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints, Oper. Res. Lett. 33 (1) (2005) 62–70.

[13] M. Grötschel, M. Jünger, Facets of the linear ordering polytope, Math. Program. 33 (1) (1985) 43–60.

[14] S. Borne, R. Grappe, M. Lacroix, The uncapacitated asymmetric traveling salesman problem with multiple stacks, in: A.R. Mahjoub, V. Markakis, I. Milis, V.T. Paschos (Eds.), Combinatorial Optimization, in: Lecture Notes in Computer Science, vol. 7422, Springer, Berlin, Heidelberg, 2012, pp. 105–116.

[15] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, R.D. Wolf, Exponential Lower Bounds for Polytopes in Combinatorial Optimization, J. ACM 62 (2) (2015) 17:1–17:23.

[16] L. Gouveia, P. Pesneau, On extended formulations for the precedence constrained asymmetric traveling salesman problem, Networks 48 (2) (2006) 77–89.

[17] L. Gouveia, J.M. Pires, The asymmetric travelling salesman problem: on generalizations of disaggregated Miller–Tucker–Zemlin constraints, Discrete Appl. Math. 112 (1–3) (2001) 129–145.

[18] A. Sassano, On the facial structure of the set covering polytope, Math. Program. 44 (1–3) (1989) 181–202.

[19] J.E. Mitchell, Branch-and-Cut Algorithms for Combinatorial Optimization Problems, Oxford University Press, 2002.

[20] G. Dantzig, R. Fulkerson, S. Johnson, Solution of a large-scale traveling-salesman problem, J. Oper. Res. Soc. Am. 2 (4) (1954) 393–410.

[21] J.X. Hao, J.B. Orlin, A faster algorithm for finding the minimum cut in a directed graph, J. Algorithms 17 (3) (1994) 424–446.

[22] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, in: Algorithms and Combinatorics, Springer-Verlag, 1988.

[23] IBM. IBM ILOG CPLEX Optimization Studio 12.5, 2012.

[24] COIN-OR project. LEMON–Library for Efficient Modeling and Optimization in Networks, 2013.

[25] J.-F. Côté, M. Gendreau, J.-Y. Potvin, Large neighborhood search for the pickup and delivery traveling salesman problem with multiple stacks, Networks 60 (1) (2012) 19–30.

[26] E. Balas, M. Fischetti, Polyhedral theory for the asymmetric traveling salesman problem, in: G. Gutin, A.P. Punnen (Eds.), The Traveling Salesman Problem and Its Variations, in: Combinatorial Optimization, vol. 12, Springer, US, 2007, pp. 117–168.