Discrete Optimization 7 (2010) 269–282

Contents lists available at ScienceDirect

Discrete Optimization

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

The Delivery Man Problem with time windows

Géraldine Heilporn[a,b], Jean-François Cordeau[a], Gilbert Laporte[b,*]

[a] Canada Research Chair in Logistics and Transportation and CIRRELT, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[b] Canada Research Chair in Distribution Management and CIRRELT, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

ARTICLE INFO

Article history:
Received 30 September 2009
Received in revised form 12 March 2010
Accepted 21 June 2010
Available online 15 July 2010

Keywords:
Delivery Man Problem
Traveling Salesman Problem
Time windows
Polyhedral analysis
Mixed integer linear programming

ABSTRACT

In this paper, a variant of the Traveling Salesman Problem with Time Windows is considered, which consists in minimizing the sum of travel durations between a depot and several customer locations. Two mixed integer linear programming formulations are presented for this problem: a classical arc flow model and a sequential assignment model. Several polyhedral results are provided for the second formulation, in the special case arising when there is a closed time window only at the depot, while open time windows are considered at all other locations. Exact and heuristic algorithms are also proposed for the problem. Computational results show that medium size instances can be solved exactly with both models, while the heuristic provides good quality solutions for medium to large size instances.

1. Introduction

The *Delivery Man Problem with Time Windows* (DMPTW) is a variant of the *Traveling Salesman Problem with Time Windows* (TSPTW) defined as follows. Let $G = (N \cup \{0\}, A)$ be a complete directed and asymmetric graph, where $N = \{1, \dots, n\}$ is a set of delivery nodes and 0 is the depot. A travel time c_{ij} is associated with each arc $(i, j) \in A$. Time windows are imposed on the beginning of service at the nodes of G: earliest and latest times are described by parameters e_i and l_i for nodes $i \in N \cup \{0\}$. If node i is reached before e_i, waiting occurs before service begins at this node. We also define the travel duration of node i as the difference between the beginning of service at node i and the beginning of service at the depot. The DMPTW consists in determining a Hamiltonian path on G, starting at the depot node 0, so as to minimize the sum of travel durations over all nodes $i \in N$ while respecting time windows. The cumulative objective function of the DMPTW is well suited to real applications involving passengers or perishable goods as, for instance, school bus routing and scheduling, the transportation of disabled people, and even some postal deliveries. Further, note that the time needed to go back to the depot is not included in the objective function. This means that we only care about the travel durations of the passengers or perishable goods. Similar problems, i.e., without a return to the depot, are referred to as 'open vehicle routing problems' (see for instance [1,2] or [3]).

The literature on the DMPTW is very limited. The only paper we are aware of is the one from Tsitsiklis [4], who presented NP-completeness results or polynomial algorithms for several special cases of the TSPTW and DMPTW. The *Delivery Man Problem* (DMP) (also called Traveling Repairman Problem), i.e., a DMPTW without time windows, was introduced by Lucena [5] who proposed an integer nonlinear model for the problem. The author derived lower bounds by Lagrangian relaxation and solved instances with up to 30 nodes using an enumerative algorithm. Fischetti et al. [6], van Eijl [7] and Méndez-Díaz et al. [8] presented several mixed integer linear programming formulations and valid inequalities for the

[*] Corresponding author.
E-mail address: gilbert@crt.umontreal.ca (G. Laporte).

1572-5286/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2010.06.002

DMP, and solved instances having between 15 and 60 nodes. The Méndez-Díaz et al. formulation with precedence variables outperforms all others in terms of relaxation quality and provides good results for instances involving up to 40 nodes. Note that the DMP is also known as the *Minimum Latency Problem* (MLP) in the scheduling literature, in which the most mentioned application concerns disk-head scheduling. Several authors have proposed approximation algorithms for the MLP (see, for instance, [9,10] or [11]). Wu et al. [12] also presented an exact algorithm combining dynamic programming and branch-and-bound, and solved instances with up to 26 nodes. However, neither the DMP nor the MLP includes time windows and the corresponding formulations cannot be adapted to the DMPTW.

The TSPTW has been more extensively studied. Baker [13] proposed a non-differentiable and non-convex model and solved instances with up to 50 nodes by branch-and-bound, using a longest path algorithm to obtain lower bounds. Langevin et al. [14] presented a mixed integer linear formulation based on a two-commodity network flow and solved instances with up to 60 nodes. However, their formulation is not well suited to include a cumulative objective function. Ascheuer et al. [15] developed valid inequalities for the TSPTW and proved several polyhedral results. In a companion paper, Ascheuer et al. [16] compared three mixed integer linear formulations for the problem. They developed a branch-and-cut algorithm for their best model which is capable of solving instances with up to 70 nodes. For the same formulation, Mak and Ernst [17] proposed new cycle breaking and infeasible path inequalities. Preliminary results have shown that these tighten the optimality gap but no further numerical results were presented. However, because the latter formulation makes use of infeasible path inequalities to model time windows, it cannot handle a cumulative objective function either.

Since both DMP and TSPTW are NP-hard, several authors have focused on heuristics. For the DMP, Lucena [5] and Bianco et al. [18] have proposed 2-exchange amd 3-exchange heuristics, respectively, the initial tour being constructed by an insertion procedure. They solved instances with up to 35 nodes. A 3-exchange heuristic, coupled with a greedy initialization procedure, was also considered by Fischetti et al. [6]. The authors solved instances with up to 60 nodes. In what concerns the TSPTW, Gendreau et al. [19] proposed an adaptation of a near-optimal TSP heuristic, and obtained good quality solutions for instances with up to 100 nodes. Wolfler Calvo [20] developed a heuristic in which a related assignment problem is first solved to minimize infeasibility with respect to time windows. The corresponding solution is then reduced to a single tour and improved by a local search procedure. He solved instances with up to 200 nodes, and obtained better results than Gendreau et al. Note that other techniques have also been used to solve both DMP and TSPTW. Bianco et al. [18] solved instances of the DMP with up to 60 nodes with dynamic programming. Dumas et al. [21] and Bianco et al. [22] solved instances of the TSPTW with between 120 and 200 nodes again using dynamic programming, while Pesant et al. [23] and Focacci et al. [24] combined constraint-programming and exact optimization methods, and solved instances with up to 40 nodes.

The aim of this paper is to analyse and solve the DMPTW. We present mixed integer linear formulations for the problem, together with exact and heuristic algorithms. We also perform a polyhedral analysis of a special case. If all time windows are closed (i.e., earliest and latest times are given for all locations), finding a feasible solution is NP-hard [25]. Then the dimension of the convex hull of feasible solutions cannot be determined, and no further polyhedral results can be derived. However, this is not the case if a closed time window is imposed only at the depot and all other nodes have an open time window, as in [15]. We perform a separate analysis of this particular case.

The remainder of this paper is organized as follows. Two formulations of the DMPTW are presented in Sections 2 and 3. The first one is a classical model involving arc flow variables, while the second is a sequential assignment model that explicitly considers the position of nodes in the Hamiltonian path. Valid inequalities and polyhedral results are developed for the new sequential assignment formulation. Exact and heuristic algorithms are proposed in Section 4. Finally, computational results are presented in Section 5, followed by conclusions in Section 6.

2. Classical arc flow formulation

According to the Öncan et al. survey [26] in which several *Asymmetric Traveling Salesman Problem* (ATSP) formulations are compared, the best models are those that include precedence variables in addition to standard arc flow variables. We adapt such a formulation presented by Gouveia and Pires [27]. Let $x_{ij} : i, j \in N \cup \{0\} (i \neq j)$ be arc flow variables, while $v_{ij} : i, j \in N (i \neq j)$ and $f_{ij}^k : i, k \in N, j \in N \cup \{0\} (i \neq j)$, represent precedence variables:

$$v_{ij} = \begin{cases} 1 & \text{if node } i \text{ precedes node } j \text{ in the Hamiltonian path,} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$f_{ij}^k = \begin{cases} 1 & \text{if arc } (i, j) \text{ appears after node } k \text{ in the Hamiltonian path,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Finally, in order to deal with the cumulative objective function, variables $t_i : i \in N \cup \{0\}$ are introduced to represent the times at which service begins at the nodes.

With this notation, the DMPTW can be modelled as follows:

$$\text{(AF-DMP) minimize} \sum_{i=1}^{n} (t_i - t_0) \tag{3}$$

subject to:

$$t_i + c_{ij} \le t_j + M_{ij}(1 - x_{ij}) \quad i \in N \cup \{0\}, j \in N (i \neq j) \tag{4}$$

$$t_0 \le t_j \quad j \in N \tag{5}$$

$$e_j \le t_j \le l_j \quad j \in N \tag{6}$$

$$e_0 \le t_0 \le l_0 \tag{7}$$

$$\sum_{i \in N \cup \{0\}} x_{ij} = 1 \quad j \in N \cup \{0\} \tag{8}$$

$$\sum_{j \in N \cup \{0\}} x_{ij} = 1 \quad i \in N \cup \{0\} \tag{9}$$

$$\sum_{i \in N \cup \{0\}} f_{ji}^k - \sum_{i \in N} f_{ij}^k = 0 \quad j, k \in N (j \neq k) \tag{10}$$

$$\sum_{i \in N \cup \{0\}} f_{ji}^j = 1 \quad j \in N \tag{11}$$

$$f_{ij}^k \le x_{ij} \quad i, j, k \in N (i \neq j) \tag{12}$$

$$\sum_{j \in N \cup \{0\}} f_{ij}^k = v_{ki} \quad i, k \in N (i \neq k) \tag{13}$$

$$\sum_{p,q \in S} x_{pq} + v_{ki} - v_{kj} \le |S| - 1 \quad i, j, k \in N (i \neq j \neq k) \; S \subset N, |S| \ge 2 : i, j \in S, k \notin S \tag{14}$$

$$f_{ij}^k \ge 0 \quad i, j, k \in N (i \neq j) \tag{15}$$

$$x_{ij} \in \{0, 1\} \quad i, j \in N \cup \{0\} (i \neq j), \tag{16}$$

where $M_{ij} : i \in N \cup \{0\}, j \in N$ are sufficiently large constants.

Constraints (4) and (5) are the *schedule compatibility inequalities*: they state that if an arc (i, j) is used, then the service time at node j is at least equal to the service time at node i plus the travel time from i to j. They also ensure that node 0 is visited before any node of N. Note that constraints (4) also eliminate subtours. Constraints (6) and (7) are the *time windows inequalities* for nodes in $N \cup \{0\}$. Constraints (8) and (9) are *arc flow inequalities*. Constraints (10) *and* (11) are *precedence flow inequalities* which ensure the flow conservation for variables f. Constraints (12) and (13) link the variables x, f and v. Finally, constraints (14) are the *generalized subpath elimination inequalities*, which prevent subpaths in G and link those with the precedence variables v. Indeed, if node k precedes node i but not node j, there cannot be a path between nodes i and j and $\sum_{p,q \in S} x_{pq} \le |S| - 2$. Otherwise if node k precedes both i and j, or if nodes i and j precedes node k, then $\sum_{p,q \in S} x_{pq} \le |S| - 1$.

Unfortunately, the schedule compatibility inequalities (4) involve "big-M" constants. As a consequence, a polyhedral study of model (AF-DMP) cannot be performed. Indeed, these constraints, which constitute an important part of the polyhedral structure of the problem, would not define facets of the convex hull of feasible solutions of the model. Further, although the formulation (AF-DMP) is intuitive, it has been shown (see, e.g., [8] or [16]) that adding time variables to an ATSP formulation is computationally expensive. For these reasons, an alternative more tractable model is presented in the next section.

3. An alternative sequential assignment formulation

We now propose a new formulation for the DMPTW, which explicitly describes the node positions in the Hamiltonian path. Such formulations have been considered in [28–30] for the *Time-Dependent TSP*, and also in [31] or [32] for *Single Machine Scheduling Problems*.

Let σ_0 be the service time at node 0, and let $\sigma_t : t = 1, \dots, n$ be the service time at the tth node of N. We introduce position variables $y_{jt} : j \in N, t = 1, \dots, n$ and transition variables $w_{ij}^t : i, j \in N (i \neq j), t = 2, \dots, n$, where

$$y_{jt} = \begin{cases} 1 & \text{if node } j \text{ is the } t\text{th node of } N \text{ in the Hamiltonian path,} \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

$$w_{ij}^t = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are respectively the } (t-1)\text{st and } t\text{th nodes of } N \\ & \text{in the Hamiltonian path,} \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

With these variables, the DMPTW can be modelled as the following mixed integer linear programming model:

$$\text{(S-DMP) minimize} \sum_{t=1}^{n} (\sigma_t - \sigma_0) \tag{19}$$

subject to:

$$\sigma_1 - \sigma_0 \geq \sum_{i,j \in N : i \neq j} c_{0j} w_{ji}^2 \tag{20}$$

$$\sigma_t - \sigma_{t-1} \geq \sum_{i,j \in N : i \neq j} c_{ij} w_{ij}^t \quad t = 2, \ldots, n \tag{21}$$

$$e_0 \leq \sigma_0 \leq l_0 \tag{22}$$

$$\sigma_1 \geq \sum_{i,j \in N : i \neq j} e_j w_{ji}^2 \tag{23}$$

$$\sigma_1 \leq \sum_{i,j \in N : i \neq j} l_j w_{ji}^2 \tag{24}$$

$$\sigma_t \geq \sum_{i,j \in N : i \neq j} e_j w_{ij}^t \quad t = 2, \ldots, n \tag{25}$$

$$\sigma_t \leq \sum_{i,j \in N : i \neq j} l_j w_{ij}^t \quad t = 2, \ldots, n \tag{26}$$

$$\sum_{j \in N} y_{jt} = 1 \quad t = 1, \ldots, n \tag{27}$$

$$\sum_{t=2}^{n} y_{jt} = 1 \quad j \in N \tag{28}$$

$$\sum_{i \in N} w_{ij}^t = y_{jt} \quad j \in N, t = 2, \ldots, n \tag{29}$$

$$\sum_{i \in N} w_{ji}^t = y_{j,t-1} \quad j \in N, t = 2, \ldots, n \tag{30}$$

$$y_{jt} \in \{0, 1\} \quad j \in N, t = 1, \ldots, n \tag{31}$$

$$w_{ij}^t \geq 0 \quad i, j \in N (i \neq j), t = 2, \ldots, n. \tag{32}$$

Constraints (20) and (21) are the schedule compatibility inequalities. Constraints (22)–(26) are the time windows inequalities. Constraints (27) and (28) are the flow inequalities: (27) impose that a node is visited in each position $t = 1, \ldots, n$, whereas (28) ensure that each node is visited once. Finally, constraints (29) and (30) link the transition and position variables w_{ij}^t and y_{jt}.

Note that, thanks to the introduction of the binary position variables y_{jt}, the transition variables w_{ij}^t can be declared as continuous as one can check that $w_{ij}^t = y_{i,t-1} y_{jt}$. Alternatively, the problem could be expressed in terms of the variables σ_t and w_{ij}^t only, which would increase the number of binary variables from $O(n^2)$ to $O(n^3)$. However, preliminary tests have shown that the previous (S-DMP) provides the best computational performance.

3.1. Polyhedral study

As mentioned in Section 1, even finding a feasible solution to the DMPTW is NP-hard (see [25]). In order to assess the strength of the formulation (S-DMP) from a theoretical viewpoint, we thus perform a polyhedral study of the particular situation arising when a closed time window is imposed only at the depot, and all other nodes have an open time window. More specifically, we show that in this particular case, most constraints of model (S-DMP) define facets of the convex hull of feasible solutions.

Let $\mathcal{P}^{\mathcal{E}} = \{(\sigma, w) : (20)–(23) \text{ and } (25), (27)–(32)\}$ denote the convex hull of feasible solutions for model (S-DMP) in the particular case described above. To avoid considering irrelevant cases, it is assumed that $e_0 < l_0$ and that the time windows have been tightened so that $e_j = \max\{e_j, e_0 + c_{0j}\}$ for all $j \in N$. Let us also define a "path-w matrix" as an incidence matrix in which each row corresponds to a Hamiltonian path and each column corresponds to a variable w_{ij}^t : $i, j \in N (i \neq j), t = 2, \ldots, n$. The next lemma provides the rank of the path-w matrix.

Lemma 1. *The rank of the path-w matrix is $n^3 - 3n^2 + 2n$.*

This result is stated in [33]. We provide a proof in the Appendix. From this lemma, we can deduce the dimension of $\mathcal{P}^{\mathcal{E}}$.

One should note that the next proofs will make use of dominated feasible solutions of $\mathcal{P}^{\mathcal{E}}$. Indeed, when all time windows are open, except the one at the depot, the service time at the depot should be set as late as possible and all other service times as early as possible. For this particular situation, the optimal service times are thus uniquely determined by the choice of a Hamiltonian path. However, we recall that our aim is to theoretically assess formulation (S-DMP). For this reason, we have decided to consider the convex hull of all feasible solutions rather that the polyhedron corresponding to the non-dominated feasible solutions.

Proposition 1. *The dimension of $\mathcal{P}^{\mathcal{E}}$ is $n^3 - 3n^2 + 3n$.*

Proof. Model (S-DMP) contains $n+1+n^2+n(n-1)^2$ variables, whereas the number of equality constraints is $2n(n-1)+2n$. However, $n-1$ of these constraints can be obtained by linear combinations of others. For instance, one has $\sum_{j\in N} y_{jn} = 1 = \sum_{i,j\in N:i\neq j} w_{ij}^n = \sum_{i\in N} y_{i,n-1} = \cdots = \sum_{i\in N} y_{i1}$. As a consequence, $dim(\mathcal{P}^{\mathcal{E}}) \leq (n+1+n^2+n(n-1)^2) - (2n(n-1)+2n-(n-1)) \leq n^3-3n^2+3n$.

One can also prove that there exist n^3-3n^2+3n+1 affinely independent points in $\mathcal{P}^{\mathcal{E}}$. In the following, the points of $\mathcal{P}^{\mathcal{E}}$ will be described by their corresponding Hamiltonian path, for instance the path $(0,1,2,\ldots,n)$, together with an assignment of variables σ_t $(t=0,\ldots,n)$.

First, the Hamiltonian path $(0,1,2,\ldots,n)$ with the assignments

$$\sigma_0 = e_0; \qquad \sigma_t = \max\{e_t, \sigma_{t-1} + c_{(t-1)t}\}, \quad t=1,\ldots,n \tag{33}$$

$$\begin{aligned}&\sigma_0 = e_0; \qquad \sigma_t = \max\{e_t, \sigma_{t-1} + c_{(t-1)t}\}, \quad t=1,\ldots,n-1;\\ &\sigma_n = \max\{e_n, \sigma_{n-1} + c_{(n-1)n}\} + \epsilon\end{aligned} \tag{34}$$

$$\begin{aligned}&\sigma_0 = e_0; \qquad \sigma_t = \max\{e_t, \sigma_{t-1} + c_{(t-1)t}\}, \quad t=1,\ldots,n-2;\\ &\sigma_t = \max\{e_t, \sigma_{t-1} + c_{(t-1)t}\} + \epsilon, \quad t=n-1,n\end{aligned} \tag{35}$$

$$\ldots$$

$$\sigma_0 = e_0; \qquad \sigma_t = \max\{e_t, \sigma_{t-1} + c_{(t-1)t}\} + \epsilon, \quad t=1,\ldots,n \tag{36}$$

$$\sigma_0 = e_0 + \epsilon; \qquad \sigma_t = \max\{e_t, \sigma_{t-1} + c_{(t-1)t}\} + \epsilon, \quad t=1,\ldots,n \tag{37}$$

yield $n+2$ affinely independent points of $\mathcal{P}^{\mathcal{E}}$.

We also know by Lemma 1 that the rank of the path-w matrix is n^3-3n^2+2n. Further, because the equality $w_{ij}^t = y_{i,t-1}y_{jt}$ holds for all $i,j \in N (i \neq j)$ and $t = 2,\ldots,n$, there exists a bijection between the assignment of variables w_{ij}^t and y_{jt}. If $(0 = \pi(0), \pi(1), \pi(2), \ldots, \pi(n))$ is a Hamiltonian path in G, the corresponding variables $\sigma_t : t = 0,\ldots,n$ can be set to $\sigma_0 = e_0$ and $\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}$ for all $t = 1,\ldots,n$. Hence each row of the path-w matrix represents a feasible solution of (S-DMP), among which n^3-3n^2+2n are affinely independent. Because the path $(0,1,2,\ldots,n)$ used in the first part of the proof is also a row of the path-w matrix, this means that $\mathcal{P}^{\mathcal{E}}$ contains $(n+2)+(n^3-3n^2+2n)-1 = n^3-3n^2+3n+1$ affinely independent points. The result follows. □

We now prove that most constraints, or strengthened constraints, of model (S-DMP) define facets of $\mathcal{P}^{\mathcal{E}}$. We first present a strengthened version of inequality (20), which defines a facet of $\mathcal{P}^{\mathcal{E}}$ under a realistic condition on the time windows.

Proposition 2. *The inequality*

$$\sigma_1 - \sigma_0 \geq \sum_{\substack{i,j\in N:i\neq j,\\ l_0+c_{0j}\geq e_j}} c_{0j} w_{ji}^2 + \sum_{\substack{i,j\in N:i\neq j,\\ l_0+c_{0j}<e_j}} (e_j - l_0) w_{ji}^2 \tag{38}$$

is valid for (S-DMP). Further, it defines a facet of $\mathcal{P}^{\mathcal{E}}$ if and only if there exists $\tilde{k} \in N$ such that $l_0 + c_{0\tilde{k}} > e_{\tilde{k}}$.

Proof. To prove that the inequality is valid, assume that $w_{ji}^2 = 1$ for some $i,j \in N$. If $l_0 + c_{0j} \geq e_j$, inequality (38) becomes $\sigma_1 - \sigma_0 \geq c_{0j}$, which is valid by (20). Otherwise, i.e., if $l_0 + c_{0j} < e_j$, inequality (38) yields $\sigma_1 - \sigma_0 \geq e_j - l_0$, which is valid by (22) and (23). Now consider a path $(0 = \pi(0), \tilde{k} = \pi(1), \pi(2), \ldots, \pi(n))$. Since $l_0 + c_{0\tilde{k}} > e_{\tilde{k}}$, the following assignments are feasible for the variables σ:

$$\sigma_0 = l_0; \qquad \sigma_1 = l_0 + c_{0\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t=2,\ldots,n \tag{39}$$

$$\begin{aligned}&\sigma_0 = l_0; \qquad \sigma_1 = l_0 + c_{0\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t=2,\ldots,n-1;\\ &\sigma_n = \max\{e_{\pi(n)}, \sigma_{\pi(n)-1} + c_{(\pi(n)-1)\pi(n)}\} + \epsilon\end{aligned} \tag{40}$$

$$\begin{aligned}&\sigma_0 = l_0; \qquad \sigma_1 = l_0 + c_{0\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t=2,\ldots,n-2;\\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t=n-1,n\end{aligned} \tag{41}$$

$$\ldots$$

$$\sigma_0 = l_0; \qquad \sigma_1 = l_0 + c_{0\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t=2,\ldots,n \tag{42}$$

$$\begin{aligned}&\sigma_0 = l_0 - \epsilon; \qquad \sigma_1 = l_0 + c_{0\tilde{k}} - \epsilon;\\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t=2,\ldots,n.\end{aligned} \tag{43}$$

This yields $n+1$ affinely independent points of $\mathcal{P}^{\mathcal{E}}$. Furthermore, the rank of the path-w incidence matrix is n^3-3n^2+2n by Lemma 1. For any path $(0=\pi(0),\pi(1),\pi(2),\ldots,\pi(n))$ such that $l_0+c_{0\pi(1)}\geq e_{\pi(1)}$ (resp. $l_0+c_{0\pi(1)}<e_{\pi(1)}$), the corresponding variables σ can be set to $\sigma_0=l_0,\sigma_1=l_0+c_{0\pi(1)}$ (resp. $\sigma_0=l_0,\sigma_1=e_{\pi(1)}$) and $\sigma_t=\max\{e_{\pi(t)},\sigma_{t-1}+c_{\pi(t-1)\pi(t)}\}$ for all $t=2,\ldots,n$.

Finally, assume that there does not exist any $k\in N$ such that $l_0+c_{0k}>e_k$. Then all points of $\mathcal{P}^{\mathcal{E}}$ satisfying (38) at equality also lie on the hyperplane $\sigma_0=l_0$ by (22). The result follows. □

Corollary 1. *Under the assumption that $l_0+c_{0j}\geq e_j$ for all $j\in N$ and provided there exists $\tilde{k}\in N$ such that $l_0+c_{0\tilde{k}}>e_{\tilde{k}}$, constraint (20) of (S-DMP) defines a facet of $\mathcal{P}^{\mathcal{E}}$.*

One can also prove that the schedule compatibility inequalities (21) and the time window inequalities (22) define facets of $\mathcal{P}^{\mathcal{E}}$.

Proposition 3. *Constraints (21) of (S-DMP) define facets of $\mathcal{P}^{\mathcal{E}}$.*

Proof. Given $\tilde{t}\in\{2,\ldots,n\}$, we prove that $\sigma_{\tilde{t}}-\sigma_{\tilde{t}-1}\geq\sum_{i,j\in N:i\neq j}c_{ij}w_{ij}^{\tilde{t}}$ is facet defining for $\mathcal{P}^{\mathcal{E}}$. First, the Hamiltonian path $(0,1,2,\ldots,n)$ with the following assignments for variables σ:

$$\begin{aligned}
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\},\quad t=1,\ldots,\tilde{t}-2,\tilde{t}+1,\ldots,n;\\
&\sigma_{\tilde{t}-1}=\max\{e_{\tilde{t}-1},\sigma_{\tilde{t}-2}+c_{(\tilde{t}-2)(\tilde{t}-1)},e_{\tilde{t}}-c_{(\tilde{t}-1)\tilde{t}}\};\qquad \sigma_{\tilde{t}}=\sigma_{\tilde{t}-1}+c_{(\tilde{t}-1)\tilde{t}} &&(44)\\
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\},\quad t=1,\ldots,\tilde{t}-2,\tilde{t}+1,\ldots,n-1;\\
&\sigma_{\tilde{t}-1}=\max\{e_{\tilde{t}-1},\sigma_{\tilde{t}-2}+c_{(\tilde{t}-2)(\tilde{t}-1)},e_{\tilde{t}}-c_{(\tilde{t}-1)\tilde{t}}\};\quad \sigma_{\tilde{t}}=\sigma_{\tilde{t}-1}+c_{(\tilde{t}-1)\tilde{t}};\\
&\sigma_n=\max\{e_n,\sigma_{n-1}+c_{(n-1)n}\}+\epsilon &&(45)\\
&\ldots\\
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\},\quad t=1,\ldots,\tilde{t}-2;\\
&\sigma_{\tilde{t}-1}=\max\{e_{\tilde{t}-1},\sigma_{\tilde{t}-2}+c_{(\tilde{t}-2)(\tilde{t}-1)},e_{\tilde{t}}-c_{(\tilde{t}-1)\tilde{t}}\};\qquad \sigma_{\tilde{t}}=\sigma_{\tilde{t}-1}+c_{(\tilde{t}-1)\tilde{t}};\\
&\sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\}+\epsilon,\quad t=\tilde{t}+1,\ldots,n &&(46)\\
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\},\quad t=1,\ldots,\tilde{t}-2;\\
&\sigma_{\tilde{t}-1}=\max\{e_{\tilde{t}-1},\sigma_{\tilde{t}-2}+c_{(\tilde{t}-2)(\tilde{t}-1)},e_{\tilde{t}}-c_{(\tilde{t}-1)\tilde{t}}\}+\epsilon;\qquad \sigma_{\tilde{t}}=\sigma_{\tilde{t}-1}+c_{(\tilde{t}-1)\tilde{t}};\\
&\sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\}+\epsilon,\quad t=\tilde{t}+1,\ldots,n &&(47)\\
&\ldots\\
&\sigma_0=e_0+\epsilon;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\}+\epsilon,\quad t=1,\ldots,\tilde{t}-2,\tilde{t}+1,\ldots,n;\\
&\sigma_{\tilde{t}-1}=\max\{e_{\tilde{t}-1},\sigma_{\tilde{t}-2}+c_{(\tilde{t}-2)(\tilde{t}-1)},e_{\tilde{t}}-c_{(\tilde{t}-1)\tilde{t}}\}+\epsilon;\qquad \sigma_{\tilde{t}}=\sigma_{\tilde{t}-1}+c_{(\tilde{t}-1)\tilde{t}} &&(48)
\end{aligned}$$

yield $n+1$ affinely independent points of $\mathcal{P}^{\mathcal{E}}$.

Next, the rank of the path-w incidence matrix is n^3-3n^2+2n by Lemma 1. For any path $(0=\pi(0),\pi(1),\pi(2),\ldots,\pi(n))$, the variables σ can be set to $\sigma_0=e_0,\sigma_t=\max\{e_{\pi(t)},\sigma_{t-1}+c_{\pi(t-1)\pi(t)}\}$ for all $t=1,\ldots,\tilde{t}-2,\tilde{t}+1,\ldots,n$, $\sigma_{\tilde{t}-1}=\max\{e_{\pi(\tilde{t}-1)},\sigma_{\tilde{t}-2}+c_{\pi(\tilde{t}-2)\pi(\tilde{t}-1)},e_{\pi(\tilde{t})}-c_{\pi(\tilde{t}-1)\pi(\tilde{t})}\}$ and $\sigma_{\tilde{t}}=\sigma_{\tilde{t}-1}+c_{\pi(\tilde{t}-1)\pi(\tilde{t})}$. □

Proposition 4. *Constraints (22) of (S-DMP) define facets of $\mathcal{P}^{\mathcal{E}}$.*

Proof. We show that $\sigma_0\geq e_0$ is facet defining for $\mathcal{P}^{\mathcal{E}}$. The proof that $\sigma_0\leq l_0$ defines a facet of $\mathcal{P}^{\mathcal{E}}$ is obtained by replacing e_0 with l_0.

First consider the Hamiltonian path $(0,1,2,\ldots,n)$ together with the corresponding variables σ:

$$\begin{aligned}
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\},\quad t=1,\ldots,n &&(49)\\
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\},\quad t=1,\ldots,n-1;\\
&\sigma_n=\max\{e_n,\sigma_{n-1}+c_{(n-1)n}\}+\epsilon &&(50)\\
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\},\quad t=1,\ldots,n-2;\\
&\sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\}+\epsilon,\quad t=n-1,n &&(51)\\
&\ldots\\
&\sigma_0=e_0;\qquad \sigma_t=\max\{e_t,\sigma_{t-1}+c_{(t-1)t}\}+\epsilon,\quad t=1,\ldots,n, &&(52)
\end{aligned}$$

which yield $n+1$ affinely independent points of $\mathcal{P}^{\mathcal{E}}$.

Next, the rank of the path-w incidence matrix is still n^3-3n^2+2n, and the variables σ can be set to $\sigma_0=e_0$ and $\sigma_t=\max\{e_{\pi(t)},\sigma_{t-1}+c_{\pi(t-1)\pi(t)}\}$ for all $t=1,\ldots,n$. The result follows. □

The time window inequality (23) also defines a facet of $\mathscr{P}^{\mathcal{E}}$ under some realistic condition on the parameters e_i ($i \in N \cup \{0\}$).

Proposition 5. *Constraints (23) of (S-DMP) is facet defining for $\mathscr{P}^{\mathcal{E}}$ if and only if there exists $\tilde{k} \in N$ such that $e_0 + c_{0\tilde{k}} < e_{\tilde{k}}$.*

Proof. Consider a path $(0 = \pi(0), \tilde{k} = \pi(1), \pi(2), \ldots, \pi(n))$ with the following assignments for variables σ:

$$\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t = 1, \ldots, n \tag{53}$$

$$\begin{aligned} &\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t = 1, \ldots, n-1; \\ &\sigma_n = \max\{e_{\pi(n)}, \sigma_{n-1} + c_{\pi(n-1)\pi(n)}\} + \epsilon \end{aligned} \tag{54}$$

$$\begin{aligned} &\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t = 1, \ldots, n-2; \\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t = n-1, n \end{aligned} \tag{55}$$

$$\ldots$$

$$\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t = 2, \ldots, n \tag{56}$$

$$\sigma_0 = e_0 + \epsilon; \qquad \sigma_1 = e_{\tilde{k}}; \qquad \sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t = 2, \ldots, n. \tag{57}$$

One obtains $n + 1$ affinely independent points of $\mathscr{P}^{\mathcal{E}}$. Next, the rank of the path-w incidence matrix is $n^3 - 3n^2 + 2n$ by Lemma 1. For any path $(0 = \pi(0), \pi(1), \pi(2), \ldots, \pi(n))$, the corresponding variables σ can be set to $\sigma_0 = e_0, \sigma_1 = e_{\pi(1)}$ and $\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}$ for all $t = 2, \ldots, n$ (indeed, recall that the time windows have been tightened so that $e_j = \max\{e_j, e_0 + c_{0j}\}$ for all $j \in N$).

To prove the result, assume by contradiction that $e_0 + c_{0k} \geq e_k$ for all $k \in N$. Then all points of $\mathscr{P}^{\mathcal{E}}$ that satisfy (23) at equality also lie on the hyperplane $\sigma_0 = e_0$ by (20). □

Finally, the time window inequality involving the second node of N in the Hamiltonian path can be strengthened. The resulting constraint also defines a facet of $\mathscr{P}^{\mathcal{E}}$.

Proposition 6. *The inequality*

$$\sigma_2 \geq \sum_{i,j \in N : i \neq j} \max\{e_i, e_j + c_{ji}\} w_{ji}^2 \tag{58}$$

defines a facet of $\mathscr{P}^{\mathcal{E}}$ if and only if there exist $\tilde{k}_1, \tilde{k}_2 \in N$ such that $e_{\tilde{k}_1} + c_{\tilde{k}_1\tilde{k}_2} < e_{\tilde{k}_2}$.

Proof. First assume that $e_j + c_{ji} \geq e_i$ for all $i, j \in N$. Then any point of $\mathscr{P}^{\mathcal{E}}$ satisfying (58) at equality also lies on the hyperplane $\sigma_1 = \sum_{i,j \in N : i \neq j} e_j w_{ji}^2$ by (21). Hence the condition stated in Proposition 6 is necessary, so that (58) is facet defining for $\mathscr{P}^{\mathcal{E}}$.

In order to prove that the assumption is also sufficient, consider a Hamiltonian path $(0 = \pi(0), \tilde{k}_1 = \pi(1), \tilde{k}_2 = \pi(2), \pi(3), \ldots, \pi(n))$. The following settings for variables σ yield $n + 1$ affinely independent points of $\mathscr{P}^{\mathcal{E}}$:

$$\begin{aligned} &\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}_1}; \qquad \sigma_2 = e_{\tilde{k}_2}; \\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t = 1, \ldots, n \end{aligned} \tag{59}$$

$$\begin{aligned} &\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}_1}; \qquad \sigma_2 = e_{\tilde{k}_2}; \\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}, \quad t = 1, \ldots, n-1; \\ &\sigma_n = \max\{e_{\pi(n)}, \sigma_{n-1} + c_{\pi(n-1)\pi(n)}\} + \epsilon \end{aligned} \tag{60}$$

$$\ldots$$

$$\begin{aligned} &\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}_1}; \qquad \sigma_2 = e_{\tilde{k}_2}; \\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t = 3, \ldots, n \end{aligned} \tag{61}$$

$$\begin{aligned} &\sigma_0 = e_0; \qquad \sigma_1 = e_{\tilde{k}_1} + \epsilon; \qquad \sigma_2 = e_{\tilde{k}_2}; \\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t = 3, \ldots, n \end{aligned} \tag{62}$$

$$\begin{aligned} &\sigma_0 = e_0 + \epsilon; \qquad \sigma_1 = e_{\tilde{k}_1} + \epsilon; \qquad \sigma_2 = e_{\tilde{k}_2}; \\ &\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\} + \epsilon, \quad t = 3, \ldots, n. \end{aligned} \tag{63}$$

Note that the condition stated in Proposition 6 ensures the feasibility of the two last assignments for variables σ. Furthermore, for any path $(0 = \pi(0), \pi(1), \pi(2), \ldots, \pi(n))$, the corresponding variables σ can be set to $\sigma_0 = e_0, \sigma_1 = e_{\pi(1)}$, $\sigma_2 = \max\{e_{\pi(2)}, e_{\pi(1)} + c_{\pi(1)\pi(2)}\}$ and $\sigma_t = \max\{e_{\pi(t)}, \sigma_{t-1} + c_{\pi(t-1)\pi(t)}\}$ for all $t = 3, \ldots, n$. As the rank of the path-w incidence matrix is $n^3 - 3n^2 + 2n$ as before, the result follows. □

Hence only the time window inequalities involving the third to the nth node of N in the Hamiltonian path do not define facets of $\mathscr{P}^{\mathcal{E}}$. This means that model (S-DMP) is strong, at least theoretically. In Section 5, the latest model will be computationally compared with the classical model (AF-DMP).

4. Algorithms

In this section, we describe both an exact and a heuristic solution method for the DMPTW.

4.1. Exact algorithm

Models (AF-DMP) and (S-DMP) can be implemented and solved exactly using a general purpose branch-and-cut algorithm. The subtour elimination constraints (14) in (AF-DMP) are separated in a classical way. Given a current solution, we create a supporting graph $G^* = (N \cup \{0\}, A^*)$, where $(i, j) \in A^*$ has a capacity equal to the value x^*_{ij} taken by x_{ij}. First, we determine the number of connected components in the graph induced by the arcs with strictly positive capacity. If there is more than one connected component, the corresponding subtour elimination constraints are appended to the model. Next, for all $i, j, k \in N$, we aggregate the nodes i, j into a node ij, and we look for the minimum capacity cut between nodes ij and k. If the corresponding subtour elimination constraint (14) is violated by the current solution, it is appended to the model.

For the DMPTW with closed time windows at all nodes, the constants M_{ij} of model (AF-DMP) are set to $M_{ij} = l_i + c_{ij} - e_j$ for all $i \in N \cup \{0\}$ and $j \in N$. If a closed time window is imposed only at the depot and all nodes of N have an open time window, then we set the constants M_{ij} as follows:

$$M_{ij} = \max_{j\in N\cup\{0\}} \{e_j\} + \sum_{k\in N:k\neq j} \max_{j\in N}\{c_{kj}\} - c_{ij} \quad i, j \in N \tag{64}$$

$$M_{0j} = \max_{j\in N\cup\{0\}} \{e_j\} + \sum_{k\in N\cup\{0\}:k\neq j} \max_{j\in N}\{c_{kj}\} - c_{0j} \quad j \in N. \tag{65}$$

Furthermore, using logical implications between the time windows $[e_i, l_i] : i \in N$ and the travel times $c_{ij} : i \in N \cup \{0\}$, $j \in N$, the time windows are tightened as follows:

$$e_j = \max\left\{e_j, \min_{i\in N\cup\{0\}:i\neq j}\{e_i + c_{ij}\}\right\} \quad j \in N \tag{66}$$

$$l_j = \min\left\{l_j, \max\left\{e_j, \max_{i\in N\cup\{0\}:i\neq j}\{l_i + c_{ij}\}\right\}\right\} \quad j \in N \tag{67}$$

$$l_0 = \min\left\{l_0, \max_{i\in N:i\neq j}\{l_i - c_{0i}\}\right\}. \tag{68}$$

We also apply a preprocessing step on the nodes of N, setting $x_{ij} = 0$ in model (AF-DMP) (resp. $w^t_{ij} = 0$ for all $t = 2, \dots, n$ in (S-DMP)) for all $i, j \in N$ such that $e_i + c_{ij} > l_j$.

Finally, a strengthened version of model (S-DMP) is considered. In the latter, the facet defining inequalities (38) and (58) are appended to (S-DMP), together with the following valid inequalities:

$$\sigma_2 \le \sum_{i,j\in N:i\neq j} \min\{l_i, \max\{e_i, l_j + c_{ji}\}\} w^2_{ji} \tag{69}$$

$$\sigma_t \ge \sum_{i,j\in N:i\neq j} \max\{e_j, e_i + c_{ij}\} w^t_{ij} \quad t = 3, \dots, n \tag{70}$$

$$\sigma_t \le \sum_{i,j\in N:i\neq j} \min\{l_j, \max\{e_j, l_i + c_{ij}\}\} w^t_{ij} \quad t = 3, \dots, n. \tag{71}$$

The validity of these inequalities can be checked using logical implications between the schedule compatibility and the time window constraints. Furthermore, note that (69) and (71) are redundant when a closed time window is imposed only at the depot.

4.2. Heuristic

We now describe a heuristic for the DMPTW. An insertion procedure is first applied to construct an initial feasible solution of the problem. An exchange procedure is then used to perturb the current solution and to improve the objective function value.

The insertion procedure works as follows. As in [20], we solve a related *Assignment Problem* (AP) while minimizing infeasibility of time windows. Because $t_j \le l_j$ for all $j \in N$ and considering the cumulative objective function of the DMPTW, the service times at nodes of N should be as small as possible. For all $i \in N \cup \{0\}, j \in N$ such that $x_{ij} = 1$, one also knows that

$$t_j \ge \max\{t_i + c_{ij}, e_j\} \ge \max\{e_i + c_{ij}, e_j\} = e_i + c_{ij} + \bar{w}_{ij}, \tag{72}$$

where $\bar{w}_{ij} = \max\{e_j - e_i - c_{ij}, 0\}$. Hence the following AP is solved:

$$\text{(AP) minimize} \sum_{i \in N \cup \{0\}, j \in N : i \neq j} (c_{ij} + \bar{w}_{ij}) x_{ij} \tag{73}$$

subject to:

$$\sum_{i \in N \cup \{0\}} x_{ij} = 1 \quad j \in N \cup \{0\} \tag{74}$$

$$\sum_{j \in N \cup \{0\}} x_{ij} = 1 \quad i \in N \cup \{0\}, \tag{75}$$

$$x_{ij} \in \{0, 1\} \quad i, j \in N \cup \{0\}. \tag{76}$$

The AP solution yields a *main path* $(0, \ldots, k)$ containing the depot 0, as well as several subpaths not containing it. The feasibility of the main path is then checked by computing the earliest times at nodes:

$$t_0 = e_0 \tag{77}$$

$$t_j = \max\{e_j, t_{j-1} + c_{(j-1)j}\} \quad j \in \{1, \ldots, k\}. \tag{78}$$

If a node $j \in N$ is infeasible with respect to its time window $[e_j, l_j]$, i.e., if $t_j > l_j$, it is removed from the main path.

Next, the subpaths are selected one at a time for insertion in the main path. At each iteration, the selected subpath S is the one corresponding to the smallest time window width $l_i - e_i : i \in N$, among those that have not been already selected. The heuristic attempts to insert S between every pair of nodes of the main path, in the order in which they appear in the main path. If there is no feasible insertion, one tries to insert S in the reverse order. If there still is no feasible insertion, one tries to insert S by decomposing the path into blocks of single nodes, i.e., the first node of S is between nodes i and $i + 1$ of the main path ($i \in \{0, \ldots, k\}$), the second node of S is between nodes $i + 1$ and $i + 2$ of the main path, etc. This process stops either as soon as a feasible insertion of S has been found, or when all the previous insertions have been considered. If an insertion is feasible, it is implemented and another subpath is selected for insertion.

When all subpaths have been considered for insertion into the main path, the related AP is solved on the nodes that do not belong to the main path. Again, the subpaths are selected one at a time for insertion in the main path. When no more feasible insertion of the subpaths exist, the remaining nodes are sorted by increasing width of time windows. The nodes are then iteratively selected for insertion between any pair of nodes of the main path, in the order in which they appear in the main path. Whenever a feasible insertion is found, it is implemented and the next node is selected. At the end of this process, if there are still nodes that cannot be inserted in the main path, a backtracking process is applied. A node is first chosen randomly and removed from the main path. Then all remaining nodes are iteratively selected for insertion in the main path. If there is still no feasible insertion, a second node is randomly chosen and removed from the main path. The process ends as soon as a feasible insertion has been identified.

This process yields a feasible Hamiltonian path $P = (0, 1, \ldots, n)$. The times at nodes of N are fully determined by the service time at the depot. Hence, in order to minimize the objective function, one should start from the depot as late as possible. As in [34], we define the forward time slack at node i for a sequence $(i, \ldots, j)$ as the largest possible delay of node i such that the corresponding sequence remains feasible, i.e.,

$$F_i^{(i,\ldots,j)} = \min_{i \leq k \leq j} \left\{ l_k - \left(t_i + \sum_{i \leq p < k} c_{p,p+1} \right) \right\}. \tag{79}$$

The latest service time at the depot such that P remains feasible is given by $F_0^{(0,1,\ldots,n)}$, which can be determined through the recursion formula

$$F_0^{(0,\ldots,i,i+1)} = \min \left\{ F_0^{(0,\ldots,i)}, l_{i+1} - t_{i+1} + \sum_{0 < p \leq t+1} W_p \right\}. \tag{80}$$

Given an initial feasible solution of the problem, an exchange procedure is used to improve the current value of the objective function. We consider 2-opt and Or-opt exchanges of nodes. A 2-opt exchange consists in replacing two arcs $(i, i + 1)$ and $(j, j + 1)$ of the current path by (i, j) and $(i + 1, j + 1)$, this also involving that the sequence $(i + 1, \ldots, j)$ is reversed in the new path. An Or-opt exchange consists in moving a sequence $(i_1, \ldots, i_2)$ of the current path between a pair of nodes $(j, j + 1)$. Such a sequence (usually of length 1, 2 or 3) can be moved forwards or backwards in the path, depending of the pair of nodes $(j, j + 1)$.

At each iteration, a lexicographic search is used to select the best feasible exchange of nodes. This implies that both a feasibility test and an optimality test are performed. For each possible exchange, the feasibility test consists in computing the forward time slack F_0 at node 0 for the new path, using the procedure described in [34]. One can then conclude that the new path is feasible if $F_0 \geq -t_0 + e_0$, where t_0 is the current time at node 0. If the new path is feasible, an optimality test is used to compute the objective function value with the new path. The time at node 0 can be set to $t_0 := t_0 + \min\{F_0, \sum_{0<p<n} W_p\}$,

Table 1
The DMPTW with a closed time window only at the depot, on TSPLIB instances.

	(AF-DMP)				(S-DMP)				Strengthened (S-DMP)			
	Solved	Gap	CPU	Nodes	Solved	Gap	CPU	Nodes	Solved	Gap	CPU	Nodes
12	3	3.17	1	490	13	20.58	27	6 613	13	17.7	7	1 213
15	3	3.11	21	7 041	11	14.61	240	16 459	13	16.41	86	6 725
20	1	3.62	1428	172 305	9	15.68	256	8 613	9	15.22	413	13 116
25	0	*	*	*	6	12	448	7 181	6	11	497	8 102

where the sum of waiting times on the new path is again calculated as in [34]. The objective function value of the new path can then be computed recursively.

Two different exchange procedures, with or without a tabu list, were developed and compared. In the procedure without a tabu list, the best feasible exchange is executed at each iteration if and only if it yields a better objective function value. In the procedure with the tabu list, non-improving moves are allowed in order to escape from local optima. However, the best known solution is always recorded. In order to avoid cycles, the last moves are stored in a tabu list and the procedure is stopped after a given number of iterations.

5. Computational results

In this section, both the classical model (AF-DMP) and the alternative model (S-DMP) are tested on numerical instances. A strengthened version of model (S-DMP) is also tested. In the latter, inequalities (38) to (71) are appended to the formulation (S-DMP). The models have been implemented in C++ and solved using ILOG CPLEX 10.1 and the Concert Library. We have not modified any CPLEX parameter setting. In particular, as our goal is to compare two formulations for the DMPTW, the CPLEX cuts have been enabled in all computational results. All tests were run on an AMD Opteron 285 computer (2.6 GHz) running Linux.

For the particular case of the DMPTW in which a closed time is imposed only at the depot and all other nodes have open time windows, two sets of instances are considered. The first set initially comes from TSPLIB [35], but the instances were adapted to fit our problem. The earliest times at nodes are randomly generated in $[0, T]$, the latest time at node 0 lying in $[e_0, T]$, where T is the average length of a Hamiltonian path starting at node 0 and going through all nodes of N. The latter is computed as n^{-1} times the sum of c_{ij} over all $i \in N \cup \{0\}, j \in N$ such that $i \neq j$. The number of nodes in the data is also reduced to generate 13 instances of each size, namely 12, 15, 20 and 25 nodes, in addition to the depot node 0. Note that all instances are available on the website http://www.hec.ca/chairedistributique/data/dmptw. The second set of instances were derived by Ascheuer [36] from a stacker crane application. These involve from 10 to 67 nodes plus the depot. They are used to compare the models (AF-DMP) and (S-DMP) for the particular case, but also for the DMPTW with closed time windows at all nodes. The latest instances can be found on the website http://ftp.zib.de/pub/mp-testdata/tsp/atsptw/index.html.

The results obtained on the first set of instances are presented in Table 1. Columns 'Solved' provide the number of instances solved to optimality within a maximum CPU time of one hour. For these instances, columns 'Gap', 'CPU' and 'Nodes' provide the optimality gaps, the CPU times (in seconds) and the number of nodes in the branch-and-cut tree. The optimality gap is defined as $100 \times \frac{Z_{lp}-Z_{opt}}{Z_{opt}}$, where Z_{lp} is the LP relaxation optimal solution value and Z_{opt} is the integer optimal solution value (or best integer solution value if the instance cannot be solved within one hour).

One can observe that model (S-DMP), especially when it is strengthened with inequalities (38), (58) and (70), enables the solution of far more instances than model (AF-DMP). However, the optimality gaps and number of nodes for the instances of size 12 and 15 solved to optimality, are larger for model (S-DMP) than for model (AF-DMP).

Table 2 provides the results for the second set of instances. An asterisk indicates that CPLEX could not find any feasible solution within one hour. From these results, one concludes that the new model (S-DMP) can only be used when combined with the strengthened inequalities (38), (58) and (70). Further, for half of the instances solved, the optimality gap obtained with the latter model is reduced to zero. However, model (AF-DMP) is able to solve one more instance than the strengthened model (S-DMP). Note that the second set of instances seems easier than the first one. A reason for this could be that the TSPLIB instances have been modified to fit the DMPTW, and truncated to obtain graphs with 12, 15, 20 and 25 nodes. Hence the Ascheuer instances, which are non-modified instances coming from a real application, are probably more structured. Also, several small instances as 'rbg016b' or 'rbg019b' seem particularly difficult to solve. We observe that these instances have much wider time windows than 'rbg016a' or 'rbg019a', which could explain why they are more difficult to solve.

The results for the DMPTW with closed time windows at all nodes are presented in Table 3, and are quite similar to those obtained with the particular case. As before, one concludes that only the strengthened version of model (S-DMP) can be used. For the latter model, the optimality gap is also zero for half the instances solved to optimality. However, model (AF-DMP) allows solving one more instance (among 20 instances in total) than the strengthened model (S-DMP).

Numerical experiments with the heuristic were also conducted on two sets of instances. First, the Ascheuer instances [36] from the stacker crane application were used to compare the solutions obtained by the heuristic with the optimal solutions determined by exact optimization (see Section 4). In Table 4, optimal solutions are provided in column 'Obj', while 'Opt' and

Table 2
The DMPTW with a closed time window only at the depot, on the Ascheuer instances.

Instance	(AF-DMP)			(S-DMP)			Strengthened (S-DMP)		
	Gap	CPU	Nodes	Gap	CPU	Nodes	Gap	CPU	Nodes
rbg010a	1.83	0.2	105	17.76	3.49	1 685	9.01	0.31	49
rbg016a	1.43	20.24	6 329	(8.92)	3600	249 072	6.86	167.03	18 644
rbg016b	4.31	109.63	37 445	(87.15)	3600	306 668	(73.23)	3600	207 927
rbg017a	0.03	2.24	173	0.03	358.77	7 859	0	0	0
rbg019a	0.46	4.5	284	(2.02)	3600	40 234	0.62	4.18	109
rbg019b	(3.94)	3600	893 997	(29.34)	3600	70 041	(22.54)	3600	127 093
rbg021.3	0.04	4.29	184	0.04	2687.14	21 412	0	0.01	0
rbg027a	0.06	33.18	709	(0.79)	3600	5 349	0	0.04	0
rbg031a	(0.65)	3600	118 289	(20.06)	3600	2 032	(7.09)	3600	8 239
rbg033a	(1.05)	3600	190 025	(20.61)	3600	580	(5.90)	3600	6 302
rbg034a	0.47	289.51	4 681	(13.17)	3600	394	1.45	562.43	870
rbg035a	(1.25)	3600	168 956	(15.26)	3600	180	(5.43)	3600	5 051
rbg038a	0.1	224.38	1 779	(14.09)	3600	72	0.44	73.19	91
rbg040a	0.04	260.31	1 576	(13.47)	3600	78	0	0.14	0
rbg041a	(6.62)	3600	116 565	(19.34)	3600	96	(1.26)	3600	2 045
rbg042a	(71.57)	3600	135 701	(60.75)	3600	1 525	(66.39)	3600	978
rbg048a	0.01	1215.16	5 911	(38.42)	3600	0	0	0.25	0
rbg049a	0.02	1844.5	9 751	(10.24)	3600	14	0	21.07	0
rbg055a	(3.07)	3600	42 200	(44.14)	3600	0	(1.22)	3600	155
rbg067a	(9.61)	3600	21 501	(34.88)	3600	0	(4.12)	3600	44

Table 3
The DMPTW with closed time windows at all nodes, on the Ascheuer instances.

Instance	(AF-DMP)			(S-DMP)			Strengthened (S-DMP)		
	Gap	CPU	Nodes	Gap	CPU	Nodes	Gap	CPU	Nodes
rbg010a	1.99	0.15	78	17.89	1.03	203	9.09	1.04	24
rbg016a	1.65	3.73	998	9.06	50.45	3 070	7.04	3.65	52
rbg016b	8.93	29.12	7 522	(79.98)	3600	157 226	74.44	1647.81	130 511
rbg017a	0.03	1.7	92	0.03	1182.49	20 505	0	0.01	0
rbg019a	0.5	0.57	6	2.05	599.22	11 301	0.64	7.34	109
rbg019b	(4.27)	3600	1270 325	(29.58)	3600	74 832	22.72	598.81	25 040
rbg021.3	0.05	2.42	104	(0.05)	3600	31 329	0	0.01	0
rbg027a	0.07	22.63	373	(3.64)	3600	5 396	0	0.05	0
rbg031a	(0.78)	3600	150 255	*	*	*	(6.75)	3600	7 555
rbg033a	(1.17)	3600	168 106	*	*	*	(5.73)	3600	4 370
rbg034a	0.51	60.77	537	*	*	*	1.47	973.11	1 021
rbg035a	2.39	1001.08	17 436	*	*	*	(8.65)	3600	3 659
rbg038a	0.1	67.3	408	*	*	*	0.45	213.22	58
rbg040a	0.04	44.34	123	*	*	*	0	4.25	0
rbg041a	(2.72)	3600	86 685	(24.29)	3600	80	(1.22)	3600	2 237
rbg042a	(76.44)	3600	80 001	(51.94)	3600	1 320	(54.53)	3600	1 200
rbg048a	0.01	571.33	1 951	*	*	*	0	0.34	0
rbg049a	0.01	1180.67	6 560	*	*	*	0	27.37	0
rbg055a	0.07	266.83	524	*	*	*	*	*	*
rbg067a	0.02	431.45	295	*	*	*	0.25	1307.36	2

'CPU' denote the best solutions and the CPU times (in seconds) found by the heuristic. Note that instances 'rbg41' and 'rbg42' were not solved to optimality, thus the best integer solutions are provided in parentheses.

The size of the tabu list and the number of iterations after which the exchange procedure is stopped were determined after preliminary tests. The exchange procedure is stopped after 200 iterations, while the size of the tabu list ranges from 20 to 50. Table 4 shows that optimal solutions are found by the tabu search heuristic for almost all instances. One can also observe that the use of a tabu mechanism yields slight improvements in terms of solution values.

Several benchmark data sets from the literature, namely Gendreau et al. [19] and Dumas et al. [21] instances, were used. These range between 20 and 200 nodes ($|N|$), with time window widths (W) between 120 and 200 time units (the original time windows were extended by 100 time units). The corresponding results are presented in Table 5. The best solutions and CPU times are given in columns 'Opt' and 'CPU', while columns '$\Delta(\%)$' provide the improvement (in terms of solution value) with respect to the best solutions found by the heuristic without the tabu list.

These results enable us to draw two main conclusions. First, the use of a tabu list yields much better solutions that the simple descent heuristic, but the CPU times increase significantly. Next, a tabu list of size 30 yields the best solutions without a substantial increase in CPU time, as the larger instances are solved in less than 10 min.

Table 4
Heuristic results on the Ascheuer instances.

Instance		Without tabu		Tabu list of 20		Tabu list of 30		Tabu list of 50	
$\|N\|$	Obj	Opt	CPU	Opt	CPU	Opt	CPU	Opt	CPU
rbg010a	6 333	6 333	0	6 333	0	6 333	0	6 333	0
rbg016a	13 705	13 705	0	13 705	0	13 705	1	13 705	0
rbg016b	5 014	5 580	0	5 014	1	5 014	0	5 014	0
rbg017a	34 973	34 973	0	34 973	0	34 973	0	34 973	1
rbg019a	18 947	18 947	0	18 947	0	18 947	1	18 947	0
rbg019b	10 517	10 545	0	10 545	1	10 545	0	10 545	1
rbg021.3	39 699	39 699	0	39 699	0	39 699	0	39 699	0
rbg027a	67 096	67 096	0	67 096	1	67 096	2	67 096	2
rbg031a	29 413	29 413	0	29 413	2	29 413	2	29 413	3
rbg033a	36 914	36 914	0	36 914	2	36 914	2	36 914	2
rbg034a	41 754	41 754	0	41 754	3	41 754	2	41 754	3
rbg035a	37 825	37 825	0	37 825	2	37 825	3	37 825	2
rbg038a	120 752	120 752	1	120 752	4	120 752	3	120 752	4
rbg040a	118 505	118 505	0	118 505	4	118 505	5	118 505	6
rbg041a	(16 507)	16 532	0	16 529	5	16 529	4	16 529	7
rbg042a	(7 603)	6 584	1	6 107	5	6 107	6	5 607	6
rbg048a	242 002	242 002	0	242 002	6	242 002	6	242 002	6
rbg049a	350 832	350 832	1	350 832	8	350 832	7	350 832	10
rbg055a	191 702	191 702	1	191 702	9	191 702	9	191 702	12
rbg067a	391 105	391 105	1	391 105	17	391 105	17	391 105	21

Table 5
Heuristic results from the Gendreau et al. and Dumas et al. instances.

Instance		Without tabu		Tabu list of 20			Tabu list of 30			Tabu list of 50		
$\|N\|$	W	Opt	CPU	Opt	Δ (%)	CPU	Opt	Δ (%)	CPU	Opt	Δ (%)	CPU
20	120	2 674	0	2 567	−3.99	0	2 535	−5.19	0	2 535	−5.19	1
20	140	1 932	0	1 908	−1.23	0	1 908	−1.23	0	1 908	−1.23	1
20	160	2 190	0	2 150	−1.82	0	2 149	−1.86	0	2 150	−1.82	1
20	180	2 085	0	2 046	−1.86	0	2 035	−2.39	1	2 037	−2.29	1
20	200	2 349	0	2 294	−2.33	0	2 294	−2.33	1	2 294	−2.33	1
40	120	7 535	0	7 509	−0.34	3	7 509	−0.34	3	7 496	−0.51	5
40	140	7 258	0	7 205	−0.72	3	7 205	−0.72	3	7 203	−0.75	4
40	160	6 892	0	6 659	−3.37	3	6 657	−3.4	3	6 657	−3.4	4
40	180	6 966	0	6 600	−5.24	3	6 583	−5.49	4	6 578	−5.56	5
40	200	6 457	0	6 408	−0.75	3	6 408	−0.75	4	6 408	−0.75	5
60	120	9 917	1	9 304	−6.17	13	9 303	−6.18	15	9 303	−6.18	20
60	140	9 734	1	9 131	−6.18	13	9 131	−6.18	16	9 131	−6.18	21
60	160	11 454	1	11 419	−0.3	10	11 422	−0.27	12	11 422	−0.27	17
60	180	10790	1	9 796	−9.2	12	9 713	−9.97	14	9 689	−10.19	18
60	200	10 925	1	10 758	−1.52	11	10 363	−5.13	13	10 315	−5.57	17
80	120	12 150	3	11 175	−8.01	31	11 122	−8.45	38	11 156	−8.17	52
80	140	16 101	2	14 185	−11.89	27	14 198	−11.81	33	14 131	−12.23	43
80	160	9 108	3	8 614	−5.41	26	8 623	−5.31	32	8 614	−5.41	43
80	180	11 625	3	11 236	−3.34	33	11 226	−3.42	41	11 222	−3.46	56
80	200	8 302	3	8 295	−0.07	28	8 295	−0.07	34	8 272	−0.35	47
100	120	22 269	6	19 351	−13.09	62	19 246	−13.56	73	19 368	−13.02	94
100	140	23 351	6	22 087	−5.4	60	22 078	−5.44	71	22 078	−5.44	93
100	160	28 970	4	27 469	−5.17	39	27 469	−5.17	46	27 368	−5.52	58
150	120	28 245	35	27 816	−1.51	226	27 816	−1.51	283	27 192	−3.72	388
150	140	27 768	37	27 544	−0.8	225	27 382	−1.38	285	27 382	−1.38	389
150	160	21 436	27	20 752	−3.18	180	20 752	−3.18	226	21 123	−1.45	308
200	120	18 010	44	17 886	−0.68	214	17 886	−0.68	266	17 886	−0.68	356
200	140	35 203	71	34 522	−1.92	434	34 410	−2.24	547	34 391	−2.3	750

Finally, note that we have also tried to solve the DMPTW by the exact solution method while providing the solution of the heuristic as an initial solution. However, this reduces neither the CPU time nor the number of nodes in the branch-and-bound tree.

6. Conclusions

We have studied a variant of the TSPTW with a cumulative objective function, which minimizes the sum of travel durations between a depot and several locations. Two mixed integer linear programming formulations were proposed for the problem: a classical arc flow and a sequential assignment model. In order to assess the theoretical strength of the latest model, a polyhedral analysis has been performed in the special case where a closed time window is imposed only at the depot, whereas open time windows are used at all other locations. We have shown that most constraints are facet defining for the corresponding convex hull of feasible solutions. Next, we have presented both exact and heuristic algorithms for the problem. Using a general purpose branch-and-cut solver, we were able to solve instances with up to 67 nodes within reasonable computational times for both models. Whereas the first model solves a few more instances than the second one, the latter yields optimality gaps of zero for half of the instances solved to optimality. The heuristic also performs well and provides good quality solutions, especially when a tabu list is used.

Acknowledgements

This work was partly funded by the Canadian Natural Sciences and Engineering Research Council under grants 227837-04 and 39682-05. This support is gratefully acknowledged. Thanks are due to the referees for their valuable comments.

Appendix. Proof of Lemma 1

Consider that variables $w_{ij}^t : i, j \in N, i \neq j, t = 2, \dots, n$ are sorted by lexicographic order on (t, i, j). Furthermore, for all $i, j \in N$ $(i \neq j)$, let π^i and π^{ij} denote permutations of the nodes in $N \setminus \{i, n-1, n\}$ and $N \setminus \{i, j, n-1, n\}$, respectively. The notations π_S^{ij} and $\pi_{\bar{S}}^{ij}$ represent node permutations in the complementary subsets S and $\bar{S}$, where $S \cup \bar{S} = N \setminus \{i, j, n-1, n\}$. One can check that the following combinations of rows of the path-w matrix are affinely independent:

$$\begin{aligned} f^{2,i,n} &= (0, i, n, n-1, \pi^i) - (0, n, i, n-1, \pi^i) \\ &= w_{in}^2 - w_{ni}^2 + w_{n(n-1)}^3 - w_{i(n-1)}^3 \quad i \in N \setminus \{n-1, n\} \end{aligned} \tag{81}$$

$$\begin{aligned} f^{2,n-1,n} &= (0, n-1, n, n-2, \pi^{n-2}) - (0, n, n-1, n-2, \pi^{n-2}) \\ &= w_{(n-1)n}^2 - w_{n(n-1)}^2 + w_{n(n-2)}^3 - w_{(n-1)(n-2)}^3 \end{aligned} \tag{82}$$

$$\begin{aligned} f^{t,i,j} &= (0, \pi_S^{ij}, i, j, n, n-1, \pi_{\bar{S}}^{ij}) - (0, \pi_S^{ij}, i, n, j, n-1, \pi_{\bar{S}}^{ij}) \\ &= w_{ij}^t - w_{in}^t + w_{jn}^{t+1} - w_{nj}^{t+1} + w_{n(n-1)}^{t+2} - w_{j(n-1)}^{t+2} \quad t = 2, \dots n-1, i, j \in N \setminus \{n-1, n\} : i \neq j \end{aligned} \tag{83}$$

$$\begin{aligned} f^{t,i,n-1} &= (0, \pi_S^{i(n-2)}, i, n-1, n, n-2, \pi_{\bar{S}}^{i(n-2)}) - (0, \pi_S^{i(n-2)}, i, n, n-1, n-2, \pi_{\bar{S}}^{i(n-2)}) \\ &= w_{i(n-1)}^t - w_{in}^t + w_{(n-1)n}^{t+1} - w_{n(n-1)}^{t+1} + w_{n(n-2)}^{t+2} - w_{(n-1)(n-2)}^{t+2} \\ &\quad t = 2, \dots, n-1, i \in N \setminus \{n-2, n-1, n\} \end{aligned} \tag{84}$$

$$\begin{aligned} f^{t,n-2,n-1} &= (0, \pi_S^{(n-2)(n-3)}, n-2, n-1, n, n-3, \pi_{\bar{S}}^{(n-2)(n-3)}) \\ &\quad - (0, \pi_S^{(n-2)(n-3)}, n-2, n, n-1, n-3, \pi_{\bar{S}}^{(n-2)(n-3)}) \\ &= w_{(n-2)(n-1)}^t - w_{(n-2)n}^t + w_{(n-1)n}^{t+1} - w_{n(n-1)}^{t+1} + w_{n(n-3)}^{t+2} - w_{(n-1)(n-3)}^{t+2} \quad t = 2, \dots, n-1 \end{aligned} \tag{85}$$

$$\begin{aligned} f^{t,n-1,i} &= (0, \pi_S^{i(n-2)}, n-1, i, n, n-2, \pi_{\bar{S}}^{i(n-2)}) - (0, \pi_S^{i(n-2)}, n-1, n, i, n-2, \pi_{\bar{S}}^{i(n-2)}) \\ &= w_{(n-1)i}^t - w_{(n-1)n}^t + w_{in}^{t+1} - w_{ni}^{t+1} + w_{n(n-2)}^{t+2} - w_{i(n-2)}^{t+2} \quad t = 2, \dots, n-1, i \in N \setminus \{n-2, n-1, n\} \end{aligned} \tag{86}$$

$$\begin{aligned} f^{t,n-1,n-2} &= (0, \pi_S^{(n-2)(n-3)}, n-1, n-2, n, n-3, \pi_{\bar{S}}^{(n-2)(n-3)}) \\ &\quad - (0, \pi_S^{(n-2)(n-3)}, n-1, n, n-2, n-3, \pi_{\bar{S}}^{(n-2)(n-3)}) \\ &= w_{(n-1)(n-2)}^t - w_{(n-1)n}^t + w_{(n-2)n}^{t+1} - w_{n(n-2)}^{t+1} + w_{n(n-3)}^{t+2} - w_{(n-2)(n-3)}^{t+2} \quad t = 2, \dots, n-1 \end{aligned} \tag{87}$$

$$\begin{aligned} f^{t,n,i} &= (0, \pi_S^{i(n-2)}, n, i, n-1, n-2, \pi_{\bar{S}}^{i(n-2)}) - (0, \pi_S^{i(n-2)}, n, n-1, i, n-2, \pi_{\bar{S}}^{i(n-2)}) \\ &= w_{ni}^t - w_{n(n-1)}^t + w_{i(n-1)}^{t+1} - w_{(n-1)i}^{t+1} + w_{(n-1)(n-2)}^{t+2} - w_{i(n-2)}^{t+2} \quad t = 2, \dots, n-1, i \in N \setminus \{n-2, n-1, n\} \end{aligned} \tag{88}$$

$$\begin{aligned} f^{t,n,n-2} &= (0, \pi_S^{(n-2)(n-3)}, n, n-2, n-1, n-3, \pi_{\bar{S}}^{(n-2)(n-3)}) \\ &\quad - (0, \pi_S^{(n-2)(n-3)}, n, n-1, n-2, n-3, \pi_{\bar{S}}^{(n-2)(n-3)}) \\ &= w_{n(n-2)}^t - w_{n(n-1)}^t + w_{(n-2)(n-1)}^{t+1} - w_{(n-1)(n-2)}^{t+1} + w_{(n-1)(n-3)}^{t+2} - w_{(n-2)(n-3)}^{t+2} \quad t = 2, \dots, n-1 \end{aligned} \tag{89}$$

$$\begin{aligned} f^{n,i,j} &= (0, \pi^{ij}, n, n-1, i, j) - (0, \pi^{ij}, n, i, j, n-1) \\ &= w_{n(n-1)}^{n-2} - w_{ni}^{n-2} + w_{(n-1)i}^{n-1} - w_{ij}^{n-1} + w_{ij}^n - w_{j(n-1)}^n \quad i, j \in N \setminus \{n-1, n\} : i \neq j \end{aligned} \tag{90}$$

$$\begin{aligned} f^{n,i,n-1} &= (0, \pi^{i(n-2)}, n-2, n, i, n-1) - (0, \pi^{i(n-2)}, n-2, n-1, i, n) \\ &= w_{(n-2)n}^{n-2} - w_{(n-2)(n-1)}^{n-2} + w_{ni}^{n-1} - w_{(n-1)i}^{n-1} + w_{i(n-1)}^n - w_{in}^n \quad i \in N \setminus \{n-2, n-1, n\} \end{aligned} \tag{91}$$

$$\begin{aligned} f^{n,n-2,n-1} &= (0,\pi^{(n-3)(n-2)},n-3,n,n-2,n-1)-(0,\pi^{(n-3)(n-2)},n-3,n-1,n-2,n) \\ &= w^{n-2}_{(n-3)n}-w^{n-2}_{(n-3)(n-1)}+w^{n-1}_{n(n-2)}-w^{n-1}_{(n-1)(n-2)}+w^{n}_{(n-2)(n-1)}-w^{n}_{(n-2)n} \end{aligned} \tag{92}$$

$$\begin{aligned} f^{n,n-1,i} &= (0,\pi^{i(n-2)},n-2,n,n-1,i)-(0,\pi^{i(n-2)},n-2,i,n-1,n) \\ &= w^{n-2}_{(n-2)n}-w^{n-2}_{(n-2)i}+w^{n-1}_{n(n-1)}-w^{n-1}_{i(n-1)}+w^{n}_{(n-1)i}-w^{n}_{(n-1)n} \quad i\in N\setminus\{n-2,n-1,n\}. \end{aligned} \tag{93}$$

Combinations (81)–(89) form an upper triangular matrix with unit determinant and are thus affinely independent. One can also check that (90)–(93) are affinely independent from all other combinations. Indeed, each combination (90) contains terms w^{n-1}_{ij}, w^{n}_{ij} and no w^{n}_{jn} nor w^{n}_{nj}, (91) (resp. (92)) contain terms $w^{n}_{i(n-1)}$, w^{n}_{in} and no $w^{n}_{n(n-1)}$, $w^{n}_{(n-1)n}$ nor $w^{n}_{(n-1)i}$ (resp. the same terms with $i=n-2$), while (93) contains terms $w^{n}_{(n-1)i}$, $w^{n}_{(n-1)n}$ and no $w^{n}_{n(n-1)}$ nor $w^{n}_{i(n-1)}$.

There are $n-1$ combinations of class $f^{2,i,n}$ or $f^{2,n-1,n}$, $(n-2)^2(n-3)$ combinations of class $f^{t,i,j}$, and $3(n-2)^2$ combinations among the classes $f^{t,i,n-1}$, $f^{t,n-2,n-1}$, $f^{t,n-1,i}$, $f^{t,n-1,n-2}$, $f^{t,n,i}$ or $f^{t,n,n-2}$. Further, there are $(n-2)(n-3)$ combinations of class $f^{n,i,j}$ and $2n-5$ combinations among the classes $f^{n,i,n-1}$, $f^{n,n-2,n-1}$ and $f^{n,n-1,i}$. The result follows. □

References

[1] F. Li, B.L. Golden, E.A. Wasil, The open vehicle routing problem: algorithms, large-scale test problems, and computational results, Computers & Operations Research 34 (2007) 2918–2930.

[2] A.N. Letchford, J. Lysgaard, R.W. Eglese, A branch-and-cut algorithm for the capacitated open vehicle routing problem, Journal of the Operational Research Society 58 (2007) 1642–1651.

[3] P.P. Repoussis, C.D. Tarantilis, G. Ioannou, The open vehicle routing problem with time windows, Journal of the Operational Research Society 58 (2007) 355–367.

[4] J.N. Tsitsiklis, Special cases of traveling salesman and repairman problems with time windows, Networks 22 (1992) 263–283.

[5] A. Lucena, Time-dependent traveling salesman problem — the deliveryman case, Networks 20 (1990) 753–763.

[6] M. Fischetti, G. Laporte, S. Martello, The delivery man problem and cumulative matroids, Operations Research 41 (1993) 1055–1064.

[7] C.A. van Eijl, A polyhedral approach to the delivery man problem. Technical Report Memorandum COSOR 95-19, Eindhoven University of Technology, The Netherlands, 1995.

[8] I. Méndez-Díaz, P. Zabala, A. Lucena, A new formulation for the traveling deliveryman problem, Discrete Applied Mathematics 156 (2008) 3223–3237.

[9] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, M. Sudan, The minimum latency problem, in: Proceedings of the 26th ACM Symposium on the Theory of Computing, 1994, pp. 163–171.

[10] M. Goemans, J. Kleinberg, An improved approximation ratio for the minimum latency problem, Mathematical Programming 82 (1998) 114–124.

[11] K. Chaudhuri, B. Godfrey, S. Rao, K. Talwar, Paths, trees and minimum latency tours, in: Proceedings of the 44th Symposium on Foundations of Computer Science, 2003, pp. 36–45.

[12] B.Y. Wu, Z.N. Huang, F.J. Zhan, Exact algorithms for the minimum latency problem, Information Processing Letters 92 (2004) 303–309.

[13] E.K. Baker, An exact algorithm for the time-constrained traveling salesman problem, Operations Research 31 (1983) 938–945.

[14] A. Langevin, M. Desrochers, J. Desrosiers, S. Gélinas, F. Soumis, A two-commodity flow formulation for the traveling salesman and makespan problems with time windows, Networks 23 (1993) 631–640.

[15] N. Ascheuer, M. Fischetti, M. Grötschel, A polyhedral study of the asymmetric travelling salesman problem with time windows, Mathematical Programming Series A 90 (2000) 475–506.

[16] N. Ascheuer, M. Fischetti, M. Grötschel, Solving the asymmetric travelling salesman problem with time windows by branch-and-cut, Networks 36 (2000) 69–79.

[17] V. Mak, A.T. Ernst, New cutting-planes for the time and/or precedence constrained ATSP and directed VRP, Mathematical Methods of Operations Research 66 (2007) 69–98.

[18] L. Bianco, A. Mingozzi, S. Ricciardelli, The traveling salesman problem with cumulative costs, Networks 23 (1993) 81–91.

[19] M. Gendreau, A. Hertz, G. Laporte, M. Stan, A generalized insertion heuristic for the traveling salesman problem with time windows, Operations Research 43 (1998) 330–335.

[20] R. Wolfler Calvo, A new heuristic for the traveling salesman problem with time windows, Transportation Science 34 (2000) 113–124.

[21] Y. Dumas, J. Desrosiers, E. Gélinas, An optimal algorithm for the traveling salesman problem with time windows, Operations Research 43 (1995) 367–371.

[22] L. Bianco, A. Mingozzi, S. Ricciardelli, Dynamic programming strateges and reduction techniques for the travelling salesman problem with time windows and precedence constraints, Operations Research 45 (1997) 365–377.

[23] G. Pesant, M. Gendreau, J.-Y. Potvin, J.-M. Rousseau, An exact constraint logic programming algorithm for the traveling salesman problem with time windows, Transportation Science 32 (1998) 12–29.

[24] F. Focacci, A. Lodi, M. Milano, A hybrid exact algorithm for the TSPTW, INFORMS Journal on Computing 14 (2002) 403–417.

[25] M.W.P. Savelsbergh, Local search for routing problems with time windows, Annals of Operations Research 4 (1985) 285–305.

[26] T. Öncan, I.K. Altinel, G. Laporte, A comparative analysis of several asymmetric traveling salesman problem formulations, Computers & Operations Research 36 (2009) 637–654.

[27] L. Gouveia, J.M. Pires, The asymmetric traveling salesman problem: on generalizations of disaggregated Miller–Tucker–Zemlin constraints, Discrete Applied Mathematics 112 (2001) 129–145.

[28] J.-C. Picard, M. Queyranne, The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling, Operations Research 26 (1978) 86–110.

[29] K.R. Fox, B. Gavish, S.C. Graves, An n-constraint formulation of the (time-dependent) traveling salesman problem, Operations Research 28 (1980) 1018–1021.

[30] L.-P. Bigras, M. Gamache, G. Savard, The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times, Discrete Optimization 5 (2008) 685–699.

[31] M. Queyranne, A.S. Schulz, Polyhedral approaches to machine scheduling. Technical Report 408, Department of Mathematics, Technical University of Berlin, 1994.

[32] A.B. Keha, K. Khowala, J.W. Fowler, Mixed integer programming formulations for single machine scheduling problems, Computers & Operations Research 36 (2009) 2122–2131.

[33] H. Abeledo, A. Pessoa, E. Uchoa, A polyhedral study of the time-dependent traveling salesman problem, in: Proceedings of the VI ALIO/EURO Workshop on Applied Combinatorial Optimization, 2008, pp. 1–6.

[34] M.W.P. Savelsbergh, The vehicle routing problem with time windows: minimizing route duration, ORSA Journal on Computing 4 (1992) 146–154.

[35] G. Reinelt, TSPLIB A traveling salesman problem library, ORSA Journal on Computing 3 (1991) 376–384.

[36] N. Ascheuer, Hamiltonian path problems in the online optimization of flexible manufacturing systems. Ph.D. Thesis, Technische Universität Berlin, 1995.