

Strengthened Formulations and Valid Inequalities for Single Delay Management in Public Transportation

Veronica Dal Sasso, Luigi de Giovanni, Martine Labbé

► **To cite this version:**

Veronica Dal Sasso, Luigi de Giovanni, Martine Labbé. Strengthened Formulations and Valid Inequalities for Single Delay Management in Public Transportation. *Transportation Science, INFORMS*, 2019, 53 (5), pp.1213-1499. 10.1287/trsc.2018.0874 . hal-01925451

HAL Id: hal-01925451

<https://hal.inria.fr/hal-01925451>

Submitted on 16 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Submitted to *Transportation Science*
manuscript (Please, provide the manuscript number!)

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Strengthened Formulations and Valid Inequalities for Single Delay Management in Public Transportation

Veronica Dal Sasso, Luigi De Giovanni*

Dipartimento di Matematica *Tullio Levi Civita* - Università degli Studi di Padova, via Trieste 63, 35121 Padova, Italy

Martine Labbé†

GOM, Université Libre de Bruxelles, Bd du Triomphe CP210/01, 1050 Bruxelles, Belgium

The Delay Management Problem arises in Public Transportation networks, often characterized by the necessity of connections between different vehicles. The attractiveness of Public Transportation networks is strongly related to the reliability of connections, which can be missed when delays or other unpredictable events occur. Given a single initial delay at one node of the network, the Delay Management Problem is to determine which vehicles have to wait for the delayed ones, with the aim of minimizing the dissatisfaction of the passengers. In this paper, we present strengthened mixed integer linear programming formulations and new families of valid inequalities. The implementation of branch-and-cut methods and tests on a benchmark of instances taken from real networks show the potential of the proposed formulations and cuts.

Key words: Delay Management; Mixed Integer Linear Programming; Polyhedral Study, Branch-and-cut

Introduction

Intermodal public transportation networks are composed of different means of transport such as trains, buses and metros. In order to encourage the use of public transports, it is essential to offer a good service. It is however impossible to provide direct routes for all origin-destination pairs, thus passengers often have to change vehicles. To minimize the inconvenience for the users, that we can associate to the total amount of time spent travelling, usually connections between vehicles of different lines are considered, and timetables are designed in such a way that the following vehicle in a connection departs shortly after the arrival of the preceding one. This choice has a drawback, that comes out when any vehicle is delayed for some reason. To overcome this issue, it is possible to

* Corresponding author, email: luigi@math.unipd.it

† The research of the third author is supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

allow some extra time in the schedule so that small delays can be recovered on the way. However, some longer delays are unavoidable, resulting in these two alternatives: either the vehicle that follows in the route waits for the delayed one (the connection is maintained), or it departs on time (the connection is dropped). In the former case the connection is not missed but the delay spreads over the network. In the latter case, only the passengers on the feeding vehicle will arrive at their destination later than planned. It can be easily understood that the best solution to minimize the dissatisfaction of passengers will be to maintain a subset of connections and to drop the remaining ones.

In this paper, we deal with a particular delay management problem as initially stated in Schöbel (2001), where a single initial delay is considered and known in advance (offline delay management), and passenger re-routing is not allowed, meaning that passengers missing a connection will wait for the next vehicle of the same line, instead of looking for alternative paths to their destinations. Moreover, vehicles in the next time period are assumed to be on time, so that the next vehicle will arrive after a known time interval, which is the same for all lines. Finally, delay propagation does not take capacity constraints into account. Under these assumptions, the *Delay Management Problem (DMP)* consists in finding the wait-depart decisions that minimize the total inconvenience over customers, measured by the sum of all passenger delays when they reach their destinations.

This problem has been formulated by Schöbel (2001) as a Mixed Integer Linear Programming (MILP) model based on an *event-activity* network representation. Further formulations and improvements are presented in Schöbel (2007). In Heilporn et al. (2008) two equivalent models are presented, that progressively reduce the number of variables. The first one, called MILP1, is based on a simplified event-activity network called *Arr-arr graph*, while the second one, called MILP2, is obtained from MILP1 by Fourier-Motzkin elimination. MILP2 reduces the number of variables at the cost of a substantial increase in the number of constraints, so that a row-generation approach is proposed to solve the model. Some improvements to MILP1 and a preliminary polyhedral study are presented in De Giovanni et al. (2014), where valid inequalities for both MILP1 and MILP2 are deduced from the results on the mixed 0-1 knapsack polytope by Marchand and Wolsey (1997), and used to strengthen the linear relaxation.

Among other works on DMP, we cite some initial study based on simulation (Suhl and Mellouli 1999, Suhl et al. 2001) or max-plus algebra (de Vries et al. 1998, Goverde 1998). The computational complexity of DMP is discussed in Gatto et al. (2005). On-line versions of the problem have been considered by Gatto et al. (2007), Berger et al. (2011) and by Adenzo-Diaz et al. (1999). In the latter the wait/depart strategy aims at minimizing the impact of delays on vehicle schedules, rather than passengers' discomfort. Further extensions to more general DMPs have been also presented in literature. The effect of network capacity restrictions in delay propagation are considered, e.g.,

in Dollevoet et al. (2015), where stations have a limited number of platforms with impact on wait/depart decisions, or in Schöbel (2009), where DMP is integrated with decisions on optimal train-to-platform reallocation. Problems arising from a limited number of tracks are addressed in Schachtebeck and Schöbel (2010). A bicriterial approach, in which both the total vehicles' delay and the number of passengers that miss a connection are minimized, is presented in Ginkel and Schöbel (2007). Another extension to DMP considers the re-routing of passengers that may choose different routes to reach their destination in case of delays (Dollevoet et al. 2010, Berger et al. 2011, Schmidt 2013). A different approach is investigated in Liebchen et al. (2010) and consists in building timetables that achieve a good trade-off between robustness against delays and increase of the total travel time. Corman et al. (2017) integrate delay management and train scheduling, and develop a model and fast heuristics to minimize the time passengers spend in the system. Schön and König (2018) introduce a stochastic version of DMP on a single line where delays are affected by uncertainty, and propose a dynamic programming approach to minimize the total passengers' delay at destination.

The aim of this paper is to improve the formulations and extend the theoretical results in De Giovanni et al. (2014), as well as to integrate them into a branch-and-cut algorithm to solve DMP. Results from previous works are summarized in Section 1, where we report the models MILP1 and MILP2 and the valid inequalities stated in De Giovanni et al. (2014). In Section 2 we improve the formulations for DMP by providing tighter values for models' coefficients and by reducing the number of variables and constraints for MILP2. In Section 3 we propose new facets for the mixed 0-1 knapsack polytope associated to each constraint of MILP2. Related inequalities represent a new class of cuts for MILP1 and MILP2 that generalizes and dominates the one proposed in De Giovanni et al. (2014). In order to verify the effectiveness of the proposed models and cuts, in addition to the linear relaxation which was briefly investigated in De Giovanni et al. (2014), two alternative branch-and-cut algorithms for DMP have been implemented, based on the improved MILP1 and MILP2: Section 4 describes exact and heuristic separation procedures, and Section 5 reports on the computational results obtained for real instances taken from literature: a first set representing a portion of the Bruxelles' intermodal public transportation network, and other instances inspired by the railway system in the Randstad, in the Netherlands (Dollevoet et al. 2015). Some final remarks and hints for future research are given in the concluding Section 6.

1. Basic formulations and valid inequalities

We summarize the two MILP models and the valid inequalities presented in Heilporn et al. (2008) and De Giovanni et al. (2014).

The transportation network is represented by an Arr-Arr graph $(K, A^R \cup A^C)$, that is an event-activity network (Elmaghraby 1977) in which each node $k \in K$ represents the arrival of one vehicle

at one station (events) and there are two types of arcs (activities): direct-ride arcs (A^R), linking the arrivals of the same vehicle at two consecutive stations of the same line, and connection arcs (A^C), linking the arrivals of two vehicles of different lines among which a connection exists (a passenger transfer activity is included). Let $slack_{ij}$ be the time that can be recovered along the arc (i, j) , that is computed as the difference between the scheduled arrival times (according to timetable) and the minimum necessary time related to the riding and/or transfer time corresponding to the arc itself. Passenger routes correspond to paths on the Arr-Arr graph, and the corresponding set is denoted by P . Each path $p \in P$ is a sequence of direct-ride and connection arcs from an origin a_p to a destination v_p . Let A_p be the set of arcs of path $p \in P$, $A_p^C \subseteq A^C$ (resp. $A_p^R \subseteq A^R$) the set of connection (resp. direct-ride) arcs in p , and c_p the number of passengers traveling on p . Finally, let T be the inter-arrival time of two vehicles of the same line at the same station, which we suppose to be the same for all the lines, and D the initial delay.

Model MILP1 considers the following decision variables:

- d_k that represents the delay at node $k \in K$,
- $z_p = \begin{cases} 1 & \text{if all connections on path } p \in P \text{ are maintained} \\ 0 & \text{otherwise,} \end{cases}$
- $u_p = \begin{cases} d_{v_p} & \text{if all connections on path } p \in P \text{ are maintained} \\ 0 & \text{otherwise.} \end{cases}$

Assuming, without loss of generality, that the initial delay occurs at node 1, the MILP1 formulation is as follows (Heilporn et al. 2008, De Giovanni et al. 2014):

$$(MILP1) \quad \min \quad \sum_{p \in P} c_p [(1 - z_p)T + u_p] \quad (1)$$

$$s.t. \quad d_1 \geq D \quad (2)$$

$$d_j \geq d_i - slack_{ij} \quad \forall (i, j) \in A^R \quad (3)$$

$$d_j \geq d_i - slack_{ij} - M_{ij}(1 - z_p) \quad \forall p \in P, (i, j) \in A_p^C \quad (4)$$

$$u_p \geq d_{v_p} - D(1 - z_p) \quad \forall p \in P \quad (5)$$

$$d_i \geq 0 \quad \forall i \in K \quad (6)$$

$$u_p \geq 0 \quad \forall p \in P \quad (7)$$

$$z_p \in \{0, 1\} \quad \forall p \in P \quad (8)$$

where $M_{ij} = D - slack_{ij}$.

The objective function (1) minimizes the total delay suffered by all passengers on all the paths. Each delay is weighted by u_p if all connections on path p are maintained, T otherwise. In fact, since we do not allow here for passenger re-routing, people missing a connection will have to wait for

the next vehicle of the same line. The initial delay is set by constraint (2). Constraints (3) describe how the delay spreads along direct-ride arcs, while constraints (4) involve connection arcs: if $z_p = 1$ they describe how the delay spreads, otherwise they are redundant (the delay does not spread). Constraints (5) set the value of u_p according to its definition. The last constraints set variables' bounds and domains.

Observing that variables d_k do not appear in the objective function, Fourier-Motzkin elimination is used to project the feasible region on the space of the variables z_p and u_p . To this end, the set $\bar{\Pi}$ of all the paths in the Arr-Arr graph is defined. Notice that $P \subset \bar{\Pi}$ and that $c_p = 0$ for all $p \in \bar{\Pi} \setminus P$. This yields the following formulation (Heilporn et al. 2008, De Giovanni et al. 2014):

$$\text{(MILP2*)} \quad \min \quad \sum_{p \in \bar{\Pi}} c_p [(1 - z_p)T + u_p] \quad (9)$$

$$\text{s.t.} \quad u_{p'} \geq Dz_{p'} - \sum_{(i,j) \in A_p} \text{slack}_{ij} - \sum_{(i,j) \in A_p^C} M_{ij}(1 - \tilde{z}_{i,j}) \quad (10)$$

$$\forall p, p' \in \bar{\Pi} : a_p = 1, v_{p'} = v_p$$

$$u_p \geq 0 \quad \forall p \in \bar{\Pi} \quad (11)$$

$$z_p \in \{0, 1\} \quad \forall p \in \bar{\Pi}. \quad (12)$$

where $\tilde{z}_{i,j} = \max_{p \in \bar{\Pi} : (i,j) \in A_p^C} z_p$. Constraints (10) allow determining the final delay on a path p' by linking it to the delay that spreads through any of the paths $p \in \bar{\Pi}$ starting at the delayed node 1 and ending in the same node as p' . In fact, these constraints state that if all connections on path p' and on at least a path for each connection in p are maintained, then the delay $u_{p'}$ at the end of path p' will be at least the initial delay minus the time that can be recovered along p (in this case, by the objective function, $u_p = u_{p'}$). If any of the connections is not maintained, these constraints are redundant.

Due to the definition of variables \tilde{z} , constraints (10) are not linear. The constraint is linearized by considering, in the last term, the sum over each possible ordered set of paths $(p''_1, p''_2 \dots p''_{|A_p^C|})$ where the t -th path must contain the t -th connection of path p . Since we add a constraint for each possible such subset of paths, the constraint corresponding to the maximum value of z for each connection is also included in the formulation, as required by (10), together with other redundant constraints. This is formalized by the following model, corresponding to MILP2 (Heilporn et al. 2008, De Giovanni et al. 2014), where $P_t^p \subseteq \bar{\Pi}$ denotes the set of all the paths containing the t -th connection arc of p , and $\alpha(p, i, j) = t$ if and only if (i, j) is the t -th connection arc of p :

$$\text{(MILP2)} \quad \min \quad \sum_{p \in \bar{\Pi}} c_p [(1 - z_p)T + u_p] \quad (13)$$

$$\text{s.t. } u_{p'} \geq Dz_{p'} - \sum_{(i,j) \in A_p} \text{slack}_{ij} - \sum_{(i,j) \in A_p^C} M_{ij} \left(1 - z_{p''_{\alpha(p,i,j)}}\right) \quad (14)$$

$$\begin{aligned} \forall p', p \in \bar{\Pi} : a_p = 1, v_p = v_{p'}, p''_t \in P_t^p, t = 1 \dots |A_p^C| \\ u_p \geq 0 \quad \forall p \in \bar{\Pi} \end{aligned} \quad (15)$$

$$z_p \in \{0, 1\} \quad \forall p \in \bar{\Pi}. \quad (16)$$

These formulations can be improved, as shown in De Giovanni et al. (2014), by special classes of cover cuts derived from the ones defined for the Mixed 0-1 Knapsack problem (Marchand and Wolsey 1997). We recall that a cover cut identifies a set of binary variables that cannot be all equal to 1 at the same time. The Mixed 0-1 Knapsack set is defined as follows:

$$Y = \{(y, s) \in \mathbb{B}^n \times \mathbb{R}_+ : \sum_{j \in J} a_j y_j \leq b + s\}. \quad (17)$$

where $J = \{1, \dots, n\}$, $a_j > 0$, $b > 0$.

A single constraint of type (14) is obtained by fixing p' and p and the combination of paths $p''_1, p''_2 \dots p''_{|A_p^C|}$. De Giovanni et al. (2014) focus on the case when $p''_i \neq p'_j$ and $p''_i \neq p'$ for each $i, j \in \{1 \dots |A_p^C|\}$, i.e., $|A_p^C| + 1$ distinct paths are involved. In this case, the associated Mixed 0-1 Knapsack set is

$$\begin{aligned} Y^{fDMP} = \left\{ (z_{p'}, z_{p''_1}, z_{p''_2} \dots z_{p''_{|A_p^C|}}, u_{p'}) \in \{0, 1\}^{|A_p^C|+1} \times \mathbb{R}_+ \text{ s.t.} \right. \\ \left. Dz_{p'} + \sum_{(i,j) \in A_p^C} M_{ij} z_{p''_{\alpha(p,i,j)}} \leq \sum_{(i,j) \in A_p} \text{slack}_{ij} + \sum_{(i,j) \in A_p^C} M_{ij} + u_{p'} \right\} \end{aligned}$$

and, among other polyhedral results on the polytope $\text{conv}(Y^{fDMP})$, the following is proven.

PROPOSITION 1 (De Giovanni et al. (2014)). *Let $p, p', p''_1, \dots, p''_{|A_p^C|}$ be paths defined as in (14) and let $p''_i \neq p'_j$ and $p''_i \neq p'$ for each $i, j \in \{1, \dots, |A_p^C|\}$. Then:*

- *inequality (14) is a facet of $\text{conv}(Y^{fDMP})$ if and only if $\text{slack}_{ij} = 0, \forall (i, j) \in A_p$.*
- *the following inequality defines a facet of $\text{conv}(Y^{fDMP})$ if and only if $D > \sum_{(i,j) \in A_p} \text{slack}_{ij}$:*

$$z_{p'} + \sum_{(i,j) \in A_p^C} z_{p''_{\alpha(p,i,j)}} \leq |A_p^C| + \frac{u_{p'}}{D - \sum_{(i,j) \in A_p} \text{slack}_{ij}}. \quad (18)$$

Inequalities (18) say that, if we consider two paths p and p' towards the same final node $v_{p'}$ and we want to keep the final delay $d_{v_{p'}} = u_{p'} = 0$, we need to drop at least a connection along p : indeed, by definition, $v_p = v_{p'}$ so that $u_p = u_{p'} = 0$; moreover, the second term on the right hand side is non negative, as we are not able to recover all the delay along p , having $D > \sum_{(i,j) \in A_p} \text{slack}_{ij}$. If $u_{p'} > 0$, in order to satisfy (18), either we drop some connection on path p' ($z_{p'} = 0$), or we still need to drop at least one connection on p , unless we consider for p' and p a large enough delay

$d_{v_{p'}} = u_{p'} \geq D - \sum_{(i,j) \in A_p} slack_{ij}$, which makes (18) redundant, being the second term of the right hand side at least 1.

Since $\text{conv}(Y^{fDMP})$ is defined by one of MILP2 model's constraints, inequalities (18) are valid for MILP2. Furthermore, they are valid for MILP1 model, as they do not involve the variables d_k and MILP2 is the projection of MILP1 on the space of the variables z_p and u_p . As a consequence, (18) can be used as cuts to strengthen both MILP1 and MILP2 formulations. De Giovanni et al. (2014) solve the linear relaxation of MILP1 by dynamically adding inequalities (18) separated by a heuristic procedure, showing sensible improvements in terms of better dual bounds and number of instances closed at the root node.

For the case when less than $|A_p^C| + 1$ distinct paths are involved by (14), De Giovanni et al. (2014) observe that inequalities (18) are still valid for the related Mixed 0-1 Knapsack set and, consequently, MILP1 and MILP2. In this paper, we will focus on this case, showing that (18) are not facet defining and introducing a class of more general facets.

2. Formulation refinements

The formulations MILP1 and MILP2 can be further improved by choosing better values for $slack_{ij}$ and M_{ij} constants and by refining the set of paths to be taken into account for constraints (10) and (14).

2.1. Improving model coefficients

Constraints (4) use the bound $M_{ij} = D - slack_{ij}$ to force the RHS to a non positive value in case at least one connection on p is dropped ($z_p = 0$), thus making the constraints redundant. In fact, we can assume that the delay at each node (variables d_k) is not greater than the initial delay D . The constraints can be tightened by computing, for each node $k \in K$, a better bound M_k for the maximum delay that we can observe at k provided that all the connections are maintained, taking advantage of the extra time allowed for each activity, represented by the parameters $slack_{ij}$ on the arcs of the Arr-arr graph. We define

$$M_k = \max\{0, D - L_{1k}\}$$

where L_{1k} is the length of the shortest path in the Arr-Arr graph from the delayed node 1 to k , using $slack_{ij}$ as arc weights. Thus, M_k represents the maximum delay that can be recorded at node k . Let us now consider an arc (i, j) : since the delay on i is bounded by M_i , and since vehicles cannot depart before the scheduled time ($d_j \geq 0$), the maximum delay that can be recovered on (i, j) is

$$\sigma_{ij} = \min\{slack_{ij}, M_i\}$$

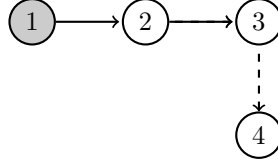


Figure 1 Example: necessary paths to correctly spread the delay.

We thus obtain the following valid formulation MILP1' for DMP:

$$\text{(MILP1')} \quad \min \sum_{p \in P} c_p [(1 - z_p)T + u_p] \quad (19)$$

$$\text{s.t. } d_1 \geq D \quad (20)$$

$$d_j \geq d_i - \sigma_{ij} \quad \forall (i, j) \in A^R \quad (21)$$

$$d_j \geq d_i - \sigma_{ij} z_p - M_i (1 - z_p) \quad \forall p \in P, (i, j) \in A_p^C \quad (22)$$

$$u_p \geq d_{v_p} - M_{v_p} (1 - z_p) \quad \forall p \in P \quad (23)$$

$$d_i \geq 0 \quad \forall i \in K \quad (24)$$

$$u_p \geq 0 \quad \forall p \in P \quad (25)$$

$$z_p \in \{0, 1\} \quad \forall p \in P \quad (26)$$

where constraints (22) and (23) replace, respectively, (4) and (5) of MILP1. Note that, again, constraints (22) and (23) are redundant if $z_p = 0$ (the RHS is non positive), while they correctly set the value of the node delays in case the initial delay spreads through maintained connections ($z_p = 1$).

2.2. Choosing the paths

Model MILP2 has been defined by Heilporn et al. (2008) after extending the set P of passenger paths to $\bar{\Pi}$, including all the possible paths on the Arr-Arr graph: this was exploited during the Fourier-Motzkin elimination steps leading to MILP2, to prove that the delay correctly spreads. We remark that MILP1 formulation, which is defined on $P \subset \bar{\Pi}$, is still valid, as the delay correctly spreads thanks to constraints (3) and (4), which directly involve the arcs of the Arr-Arr graph. On the contrary, using P instead of $\bar{\Pi}$ in MILP2 is not valid, as shown by the following example.

Let us consider the Arr-Arr graph in Figure 1, with four nodes 1, 2, 3, and 4, two direct-ride arcs (1,2) and (2,3), and a connection arc (3,4). Let 1 be the delayed node and let P contain only the path 2-3-4. Since no path starts at the delayed node, the feasible region of MILP2 would result in

simply the non-negativity constraints: the optimal solution would set z variable to 1 with objective value 0, which denotes that the delay does not correctly spread.

The example suggests that additional paths starting at the delayed node have to be taken into account. Instead of using the set $\bar{\Pi}$ of all the paths as in Heilporn et al. (2008), we define a smaller set as follows. Let \hat{p}_v be the path from the delayed node 1 to node v without connection arcs (i.e. $A_{\hat{p}_v}^C = \emptyset$). Note that \hat{p}_v may not exist and, in case it exists, it is unique. Let Θ_v be defined as follows

$$\Theta_v = \begin{cases} \sum_{(i,j) \in \hat{p}_v} \sigma_{ij} & \text{if } \hat{p}_v \text{ exists} \\ +\infty & \text{otherwise} \end{cases}$$

We define, for each node $k \in K$, the following subset:

$$\Pi_k = \{\hat{p}_k\} \cup \{q \in \bar{\Pi} \mid a_q = 1 \wedge v_q = k \wedge A_q^C \subseteq A_P^C \wedge \sum_{(i,j) \in A_q^R} \sigma_{ij} < \Theta_k\} \quad (27)$$

where $A_P^C = \bigcup_{p \in P} A_p^C$ is the set of all connection arcs included in at least one passenger path.

Each Π_k includes a subset of relevant paths starting at the delayed node, ending at some actual passenger's destination node k and propagating a large-enough delay: the following proposition shows that a valid formulation for DMP can be obtained by writing constraints (10) or (14) with reference to the sets Π_k , instead of $\bar{\Pi}$.

PROPOSITION 2. *The following formulation MILP2' is valid for DMP and the value of the linear relaxation is the same as MILP1':*

$$(MILP2') \quad \min \sum_{p \in P} c_p [(1 - z_p)T + u_p] \quad (28)$$

$$\begin{aligned} s.t. \quad u_{p'} &\geq D - \sum_{(i,j) \in A_p^R} \sigma_{ij} - \sum_{(i,j) \in A_p^C} \left[\sigma_{ij} z_{p''_{\alpha(p,i,j)}} + M_i (1 - z_{p''_{\alpha(p,i,j)}}) \right] - M_{v_{p'}} (1 - z_{p'}) \\ &\forall p' \in P, p \in \Pi_{v_{p'}}, (p''_1, p''_2, \dots, p''_{|A_p^C|}) : p''_1 \in P_1^p, p''_2 \in P_2^p \dots p''_{|A_p^C|} \in P_{|A_p^C|}^p \end{aligned} \quad (29)$$

$$u_p \geq 0 \quad \forall p \in P \quad (30)$$

$$z_p \in \{0, 1\} \quad \forall p \in P. \quad (31)$$

The proof is quite technical and it is reported in Appendix A: it reviews the steps of the Fourier-Motzkin elimination provided by Heilporn et al. (2008) in the light of making minimal the set of paths that guarantees that the initial delay correctly spreads through the Arr-Arr graph, which leads to definition (27).

3. Strengthened valid inequalities

Formulations MILP1' and MILP2' can be tightened using the following inequalities:

$$z_{p'} + \sum_{(i,j) \in A_p^C} z_{p''_{\alpha(p,i,j)}} \leq |A_p^C| + \frac{u_{p'}}{D - \sum_{(i,j) \in A_p} \sigma_{ij}}. \quad (32)$$

They directly correspond to (18) and, by the same argument used in De Giovanni et al. (2014) to prove Proposition 1, they represent facets of the mixed 0-1 knapsack set associated to each constraint (29) under the conditions $D > \sum_{(i,j) \in A_p} \sigma_{ij}$, and, for each $i, j \in \{1, \dots, |A_p^C|\}$, $p''_i \neq p''_j$ and $p''_i \neq p'$. In this case, (4) involves $|A_p^C| + 1$ distinct variables, that is, the paths selected for each connection of p are different from each others and from p' .

In this section, we will propose a new class of valid inequalities that dominates (32) in the more general case when just the condition $D > \sum_{(i,j) \in A_p} \sigma_{ij}$ holds, that is, constraints (29) involve $|A_p^C| + 1$ distinct variables or less. We demonstrate this case by the following example.

We consider a path p starting at the delayed node 1 and containing two connection arcs (a, b) and (c, d) , such that $\alpha(p, a, b) = 1$ and $\alpha(p, c, d) = 2$. Since $p \in P_1^p \cap P_2^p$, by considering $p' = p$ and the combination with $p'_1 = p'_2 = p$, one of the constraints (29) of MILP2' is given by

$$u_p \geq D - M_{v_p}(1 - z_p) - \sum_{(i,j) \in A_p^R} \sigma_{ij} - M_a - M_c + (M_a - \sigma_{ab})z_p + (M_c - \sigma_{cd})z_p$$

that is

$$(M_{v_p} + M_a - \sigma_{ab} + M_c - \sigma_{cd})z_p \leq \sum_{(i,j) \in A_p^R} \sigma_{ij} + (M_{v_p} + M_a + M_c) - D + u_p$$

The corresponding cut of type (32) is

$$z_p + z_p + z_p \leq 2 + \frac{u_p}{D - \sum_{(i,j) \in A_p} \sigma_{ij}}.$$

which is valid for MILP1' and MILP2'. However, we will see in this section that the cut is dominated by the inequality:

$$z_p \leq \frac{u_p}{D - \sum_{(i,j) \in A_p} \sigma_{ij}}.$$

This scenario occurs every time $p' \in P_i^p$, or $P_i^p \cap P_j^p \neq \emptyset$, for some $i, j = 1 \dots |A_p^C|, i \neq j$.

Let us fix one constraint of type (29) and consider the related paths $p' \in P$, $p \in \Pi_{v_p}$ and tuple $(z_{p'_1}, z_{p'_2}, \dots, z_{p'_{|A_p^C|}})$, where $p'_t \in P_t^p, \forall t = 1, \dots, |A_p^C|$. Let W be the set of distinct passenger paths p'' whose associated variable $z_{p''}$ appears in the fixed constraint (29), path p' excluded. We denote with $w_1, \dots, w_{|W|}$ the elements of W . We can define the following Mixed 0-1 Knapsack set

$$Y^{fDMP'} = \left\{ (z_{p'}, z_{w_1}, z_{w_2}, \dots, z_{w_{|W|}}, u_{p'}) \in \{0, 1\}^{|W|+1} \times \mathbb{R}_+, \left| \begin{aligned} \sum_{w \in W \cup \{p'\}} \gamma_w z_w &\leq \sum_{(i,j) \in A_p^R} \sigma_{ij} + \sum_{(i,j) \in A_p^C} M_i + M_{v_{p'}} - D + u_{p'} \end{aligned} \right. \right\} \quad (33)$$

where coefficients γ_w are defined as the sum of the coefficients in (29) related to the same variable z_w . More formally, recalling the definition of $\alpha(p, i, j)$, we denote, for each $w \in W \cup \{p'\}$, the set $A_p^C(w) = \{(i, j) \in A_p^C \mid p''_{\alpha(p, i, j)} = w\}$, and we have

$$\begin{aligned}\gamma_{p'} &= \sum_{(i, j) \in A_p^C(p')} (M_i - \sigma_{ij}) + M_{v_{p'}} \\ \gamma_w &= \sum_{(i, j) \in A_p^C(w)} (M_i - \sigma_{ij})\end{aligned}\quad \forall w \in W.$$

The mixed cover cut related to (33) is

$$z_{p'} + \sum_{w \in W} z_w \leq |W| + \frac{u_{p'}}{D - \sum_{(i, j) \in A_p} \sigma_{ij}} \quad (34)$$

It reduces to (32) if $|W| = |A_p^C|$, which is the case studied by De Giovanni et al. (2014). In the following, we will prove that (34) is facet defining for $\text{conv}(Y^{fDMP'})$. The proof is based on the following lemmata, whose rather technical proofs can be found in Appendix B.

LEMMA 1. *For any path $p \in \bar{\Pi}$ such that $a_p = 1$, $M_{v_p} \geq D - \sum_{(i, j) \in A_p} \sigma_{ij}$.*

LEMMA 2. *Assuming that $D > \sum_{(i, j) \in A_p} \sigma_{ij}$, $W \cup \{p'\}$ is a minimal cover of $Y^{fDMP'}$ and the inequality*

$$z_{p'} + \sum_{w \in W} z_w \leq |W| \quad (35)$$

defines a facet of $\text{conv}(Y^{fDMP'}|_{u_{p'}=0})$.

LEMMA 3. *If $W \neq \emptyset$, for any $w \in W \cup \{p'\}$, $\gamma_w \leq \sum_{(i, j) \in A_p^R} \sigma_{ij} + \sum_{(i, j) \in A_p^C} M_i + M_{v_{p'}} - D$.*

By Lemmata 1, 2 and 3, we can exploit the following result related to general Mixed 0-1 Knapsack sets Y defined in (17).

PROPOSITION 3 (Marchand and Wolsey (1997)). *Let $a_j \leq b$, $\forall j \in J$. If $\sum_{j \in J} \pi_j y_j \leq \pi_0$ defines a facet of $\text{conv}(Y|_{s=0})$, having $\pi \geq 0$ and $\sum_{j \in J} \pi_j > \pi_0 > 0$, then the inequality*

$$\sum_{j \in J} \pi_j y_j \leq \pi_0 + \frac{s}{\beta}$$

defines a facet of $\text{conv}(Y)$, with

$$\beta = \min_{s > 0} \frac{s}{\eta(s) - \pi_0}, \quad \eta(s) = \max \left\{ \sum_{j \in J} \pi_j y_j \mid \sum_{j \in J} a_j y_j \leq b + s, y \in \{0, 1\}^{|J|} \right\}.$$

PROPOSITION 4. *Assuming that $D > \sum_{(i, j) \in A_p} \sigma_{ij}$, inequality (34) defines a facet of $\text{conv}(Y^{fDMP'})$.*

Proof. We distinguish two cases: (i) $W \neq \emptyset$ and (ii) $W = \emptyset$.

In case (i), by Lemma 3 and being $|W| > 0$, the conditions of Proposition 3 hold. Hence we have to determine

$$\begin{aligned} \beta = & \min_{s>0} && \frac{s}{\eta(s) - |W|} \\ \eta(s) = & \max && z_{p'} + \sum_{w \in W} z_w \\ \text{s.t.} & && \sum_{w \in W \cup \{p'\}} \gamma_w z_w \leq M_{v_p} - D + \sum_{(i,j) \in A_p} \sigma_{ij} + \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) + s \\ & && z_w \in \{0, 1\} \quad \forall w \in W \cup \{p'\}. \end{aligned} \quad (36)$$

If $0 < s < D - \sum_{(i,j) \in A_p} \sigma_{ij}$, then the RHS of (36) is strictly bounded from above by

$$M_{v_p} + \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) = \sum_{w \in W \cup \{p'\}} \gamma_w.$$

This means that we cannot consider all variables z_w equal to 1, that is, $\eta(s) < |W| + 1$. Also, by Lemma 2, $W \cup \{p'\}$ is a minimal cover for $Y_0^{fDMP'}$ and, since we are interested in $s > 0$, we have $\eta(s) = |W|$.

If $s \geq D - \sum_{(i,j) \in A_p} \sigma_{ij}$, then the RHS of (36) is bounded from below by

$$M_{v_p} + \sum_{(i,j) \in A_p} (M_i - \sigma_{ij}) = \sum_{w \in W \cup \{p'\}} \gamma_w$$

so that we can consider all z_w variables equal to 1 and $\eta(s) = |W| + 1$.

Summarizing

$$\eta(s) = \begin{cases} |W| & \text{if } 0 < s < D - \sum_{(i,j) \in A_p} \sigma_{ij} \\ |W| + 1 & \text{if } s \geq D - \sum_{(i,j) \in A_p} \sigma_{ij} \end{cases}$$

and the minimum value β is obtained for $s = D - \sum_{(i,j) \in A_p} \sigma_{ij}$, that is, $\beta = D - \sum_{(i,j) \in A_p} \sigma_{ij}$.

Let us now consider case (ii). In this case, and since $D > \sum_{(i,j) \in A_p} \sigma_{ij}$, inequality (34) becomes

$$u_{p'} \geq \left(D - \sum_{(i,j) \in A_p} \sigma_{ij} \right) z_{p'}.$$

It is valid for $Y^{fDMP'}$, since it always holds if $z_{p'} = 0$ and, if $z_{p'} = 1$, noticing that in case (ii) $A_p^C(p') = A_p^C$, it reduces to the inequality that defines $Y^{fDMP'}$ in (33). Moreover, $Y^{fDMP'}$ is in \mathbb{R}^2 and, since the two affinely independent points $z'_p = u_{p'} = 0$ and $z_{p'} = 1$, $u_{p'} = D - \sum_{(i,j) \in A_p} \sigma_{ij}$ are tight, (34) defines a facet. \square

We remark here that inequality (32) is still valid for $Y^{fDMP'}$ and reduces to (34) in the special case $|W| = |A_p^C|$. In the case $|W| < |A_p^C|$, (32) are not facet defining, since they can be obtained by combining the corresponding inequality (34) with suitably chosen valid inequalities of type $z_w \leq 1$.

We have already observed that inequalities (18) can be used as cuts to strengthen both formulations MILP1 and MILP2: by similar arguments, under the condition $D > \sum_{(i,j) \in A_p} \sigma_{ij}$, inequalities (34) are valid cuts for MILP1' and MILP2'.

4. A branch-and-cut algorithm

Models MILP1, MILP2, MILP1' and MILP2' are solved with a branch-and-cut algorithm, that is, a branch-and-bound algorithm where additional cuts are used to strengthen the linear relaxation. In particular, we solve the linear relaxation and, in case the optimal solution has binary variables at a fractional value, we recursively branch by creating two subproblems where one of these variables takes either value 0 or 1. The value of the linear relaxation provides a lower bound that can be used to stop branching whenever the bound is larger than the incumbent integer-feasible solution.

MILP1 and MILP1' can be directly handled by on-the-shelf solvers, whereas MILP2 and MILP2' present a huge number of constraints (14) and (29) respectively, which suggests a row generation approach to solve their linear relaxation. We start from a partial model with no constraints (14) or (29), but the one corresponding to the trivial path starting and ending at the delayed node, and we iterate through the following steps:

1. solve the linear relaxation of the partial model;
2. determine one (or more) constraints (14) or (29) violated by the current solution;
3. if no violated constraints exist, stop;
4. otherwise add them to the partial model and iterate from Step 1.

Step 2 is performed by the constraint separation procedure.

As observed above, cuts (18), resp. (34), can be used to strengthen the linear relaxation of MILP1 and MILP2, resp. MILP1' and MILP2', to obtain tighter lower bounds for each subproblem in the branch-and-bound procedure: this gives rise to the branch-and-cut approach we propose to solve the models. However, the number of the proposed cuts is huge, and it is convenient, from a computational point of view, to generate and add them dynamically. To this end, cut separation procedures are required. Given a solution of the linear relaxation including just a subset of cuts (18) or (34), a cut separation procedure determines one or more further cuts violated by the solution itself: they are added to the linear relaxation and the process iterates. Notice that, for MILP2 and MILP2', both constraint and cut separation procedures are used to dynamically add inequalities to the partial models. In the following, we present constraint and cut separation procedures with reference to models MILP1' and MILP2', with improved coefficients: the corresponding procedures for MILP1 and MILP2 are obtained by substituting σ_{ij} with $slack_{i,j}$.

4.1. Constraint separation

As detailed in Appendix A, constraints (29) of MILP2' come from linearizing inequalities

$$u_{p'} \geq D - \sum_{(i,j) \in A_p} l_{ij} - M_{v_{p'}}(1 - z_{p'}) \quad \forall p' \in P, p \in \Pi_{v_{p'}}, \quad (37)$$

which link the delay at the end of a passenger path p' to the delay propagated on a path p from 1 to $v_p = v_{p'}$. Inequalities (37) use the terms l_{ij} , representing the amount of delay that can be saved on arc (i, j) , computed as a non-linear function of the z variables:

$$l_{ij} = \begin{cases} \min_{p \in P: (i,j) \in A_p^C} \{\sigma_{ij} z_p + M_i(1 - z_p)\} & \text{if } (i, j) \in A_p^C \\ \sigma_{ij} & \text{if } (i, j) \in A^R \end{cases}. \quad (38)$$

Since in the separation procedure the solution to the current partial MILP2' is available, we can use the values related to variables z to compute the terms l_{ij} and directly separate (37): if such a violated constraint exists, a violating constraint (29) is determined by the same pair of paths p and p' , and by the paths obtained choosing, for each connection arc $(i, j) \in A_p^C$, $p''_{\alpha(p,i,j)} = \arg \max_{p \in P: (i,j) \in A_p^C} z_p$, according to (38); otherwise, no constraint (29) is violated.

Given the optimal solution (z^*, u^*) to the linear relaxation of the current partial MILP2', we want to determine two paths p and p' such that (37), and hence (29), is violated, that is, $\sum_{(i,j) \in A_p} l_{ij} < D - u_{p'}^* - M_{v_{p'}}(1 - z_{p'}^*)$. Noticing that the right term depends only on p' , once p' is fixed the maximum constraint violation is obtained by selecting the shortest path $p \in \Pi_{v_{p'}}$ with l_{ij} as arc weights: this is done by following Procedure A.

Procedure A:

1. For each $(i, j) \in A^R \cup A_p^C$, compute l_{ij} according to (38); for each $(i, j) \in A^C \setminus A_p^C$, set $l_{ij} = +\infty$
2. For each node $k \in \bigcup_{q \in P} \{v_q\}$, determine $\tilde{d}(k) = \min_{p \in \Pi_k} \sum_{(i,j) \in A_p} l_{ij}$; let $\tilde{p}(k)$ be the related path
3. For each path $p' \in P$:
4. if $\tilde{d}(v_{p'}) > u_{p'}^* + M_{v_{p'}}(1 - z_{p'}^*)$, then inequality (37) is violated for the paths p' , $p = \tilde{p}(v_{p'})$.

Step 1 computes the terms l_{ij} corresponding to the solution at hand, setting to an infinite cost the connection arcs not included in any passenger path (we are not interested in paths including any such arcs, since the delay is not propagated on them). Step 2 computes the shortest path on the Arr-Arr graph from the delayed node 1 to any possible passenger path destination, according to the l_{ij} metric. We notice that this corresponds to propagating the initial delay and determining, for each destination node k , the path propagating the maximum delay, which is at least the one on the direct path \hat{p}_k . As a consequence, $\tilde{p}(k) \in \Pi_k$, since, by (27), all paths from 1 to k not included in Π_k propagate a delay not greater than the one on \hat{p}_k .

Observing that the Arr-Arr graph is acyclic, Step 2 can be done by a breadth first visit, and the overall computational complexity of Procedure A is $O(|A^C||P| + |A^R| + |A^C \cup A^R| + |P|) = O(|A^C||P| + |A^C \cup A^R|)$, hence polynomial in the length of the DMP input.

4.2. Cut separation

Given a fractional solution (z^*, u^*) of the LP relaxation of MILP1' or MILP2', we want to determine if it violates any of the inequalities (34) that is, we search for paths $p' \in P$, $p \in \Pi_{v_{p'}}$ and a suitable set $W \subseteq P \setminus p'$ made of paths including the connection arcs of p , such that

$$|W| - \sum_{w \in W} z_w^* < z_{p'}^* - \frac{u_{p'}^*}{D - \sum_{(i,j) \in A_p} \sigma_{ij}}. \quad (39)$$

In the following, we present cut separation procedures that fix the pair of paths p' and p , where p starts at the delayed node and $v_p = v_{p'}$, and determine the set W minimizing the LHS of (39), which corresponds to the maximum cut violation.

We remark that inequalities (34) have been derived under the condition $D > \sum_{(i,j) \in A_p} \sigma_{ij}$, as otherwise they are not valid for $Y^{fDMP'}$: e.g., the violating point $z_{p'} = z_w = 1, \forall w \in W$, $u_{p'} = 0$ belongs to $Y^{fDMP'}$, as can be easily checked. As a consequence, paths p with $D \leq \sum_{(i,j) \in A_p} \sigma_{ij}$ are not taken into account.

Preliminary checks. Before applying a cut separation procedure to a pair of paths $p' \in P$, $p \in \Pi_{v_{p'}}$, we can check the following conditions under which no violated cuts involving p' and p exist. We present this checking by increasing order of computational complexity.

1. $A_p^C = \emptyset$: under this condition, inequality (34) defines a facet of $Y^{fDMP'}$, but it is redundant for the optimal solutions of relaxed MILP1' and MILP2', for any $p' \in P$. In fact, in this case, the constraints of the two formulations imply $u_p^* \geq D - \sum_{(i,j) \in A_p} \sigma_{ij}$ and, from optimality, $u_{p'}^* = u_p^*$. It follows that $z_{p'}^* - \frac{u_{p'}^*}{D - \sum_{(i,j) \in A_p} \sigma_{ij}} \leq 0$ and (39) cannot hold, as $|W| - \sum_{w \in W} z_w^* \geq 0$.

2. $u_{p'}^* \geq M_{v_{p'}} z_{p'}^*$: in fact, since $M_{v_{p'}} = M_{v_p} \geq D - \sum_{(i,j) \in A_p} \sigma_{ij}$, in this case, we have

$$z_{p'}^* - \frac{u_{p'}^*}{D - \sum_{(i,j) \in A_p} \sigma_{ij}} \leq z_{p'}^* - \frac{u_{p'}^*}{M_{v_{p'}}} \leq z_{p'}^* - z_{p'}^* = 0$$

and (39) cannot hold. Notice the this condition includes the case $z_{p'}^* = 0$.

3. $u_{p'}^* \geq \left(D - \sum_{(i,j) \in A_p} \sigma_{ij}\right) z_{p'}^*$: as above, in this case we have $z_{p'}^* - \frac{u_{p'}^*}{D - \sum_{(i,j) \in A_p} \sigma_{ij}} \leq 0$, guaranteeing that no cut is violated.

If none of the above conditions holds, the following procedures allow separating inequalities (34) for a given pair of paths $p' \in P$, $p \in \Pi_{v_{p'}}$.

Set covering approach. The problem of finding the maximum cut violation, that is, for fixed p and p' , minimizing the LHS of (39), can be solved by the following set-covering formulation:

$$\begin{aligned} \min \quad & \sum_{w \in P \setminus \{p'\}} (1 - z_w^*) y_w \\ \text{s.t.} \quad & \sum_{w \in P: (i,j) \in A_w^C} y_w \geq 1 & \forall (i,j) \in A_p^C \\ & y_w \in \{0, 1\} & \forall w \in P \end{aligned}$$

where y_w are the variables that characterize the set W . If the value of the optimal solution is strictly negative, the related W , together with p' and p , identifies a violated cut, otherwise the current solution (z^*, u^*) violates no cuts. We remark that p' is not included in the objective function (no associated penalty), but it is included in the constraints (it can be used to cover the connection arcs), so that if $p = p'$, the model has an optimal solution with all variables set to 0 but $y_{p'} = 1$. In fact, if $z_{p'}^* > \frac{u_{p'}^*}{D - \sum_{(i,j) \in A_p} \sigma_{ij}}$, a violated inequality (34) exists with $W = \emptyset$.

Solving the set covering formulation is computationally expensive and we propose the following heuristic procedure HP1, based on the well-known approach by Chvátal (1979):

Procedure HP1:

1. Set $W = \emptyset$, $\bar{A} = A_p^C$
2. Set $\bar{w} = \arg \max_{w \in P \setminus \{p'\}} \frac{|\bar{A} \cap A_w^C|}{1 - z_w^*}$
3. Update $W = W \cup \{\bar{w}\}$, and $\bar{A} = \bar{A} \setminus A_{\bar{w}}^C$
4. If $\bar{A} = \emptyset$ then STOP, otherwise iterate from Step 2.

We start from an empty W and we iteratively add to W the path $w \in P \setminus \{p'\}$ maximizing the score $\frac{\bar{n}}{1 - z_w^*}$, where \bar{n} is the number of connection arcs of p included in w and not yet covered by other paths in W . The procedure stops as soon as all connection arcs of p are covered. Since the output W may be not optimal, the procedure may fail in identifying a violated cut, even if it exists, that is, the cut separation procedure HP1 is heuristic.

A further heuristic solution to the set covering problem comes from the observation that, given $p' \in P$ and $p \in \Pi_{v_{p'}}$, it is easy to separate inequalities (32), since the maximum violation comes from setting $p''_{\alpha(p,i,j)} = \arg \max_{p \in P: (i,j) \in A_p^C} z_p^*$. We can thus consider W as the set of involved path p'' (different from p'), and check if the corresponding (34) is violated. To perform this step we can apply the same separation procedure used in De Giovanni et al. (2014), based on enumerating all possible pairs (p', p) and checking the corresponding inequality.

We denote this procedure as **Procedure HP2** and we remark that it determines the inequality that maximizes the violation of (32), whereas the corresponding W is a heuristic solution to the set covering problem and may not maximize the violation of (34). As a consequence, Procedure HP2 is heuristic, as shown by the following example. Let p' and p be paths such that $z_{p'}^* = 0.3$, $u_{p'}^* = 0$, p includes 3 connection arcs (a, b) , (c, d) and (e, f) . Suppose we have two paths q and r with $A_q^C = \{(a, b), (c, d)\}$ and $A_r^C = \{(a, b), (c, d), (e, f)\}$. Let $z_q^* = 0.85$ and let $z_r^* = 0.8$. In this case, the maximum violation for (32) yields the inequality $z_{p'} + z_q + z_r \leq 3 + 0$, and then the inequality (34) $z_{p'} + z_q + z_r \leq 2 + 0$ (corresponding to $W = \{q, r\}$) would be checked and no violated inequality would be found. Actually, the optimal solution to the set covering problem is $W = \{r\}$, yielding the inequality $z_{p'} + z_r \leq 1 + 0$, which is violated.

Table 1 Branch-and-cut algorithms for DMP.

Algorithm:	MILP1	MILP2	MILP1'	MILP2'	MILP1'+HP1	MILP1'+HP2
Model	MILP1	MILP2	MILP1'	MILP2'	MILP1'	MILP1'
Constraints / Procedure	–	(14) / A	–	(29) / A	–	–
Cuts / Procedure	–	–	–	–	(34) / HP1	(34) / HP2

– : not used

5. Computational results

The branch-and-cut approach described above has been implemented using the solver Cplex 12.4 (IBM 2014) and Python 2.7 (and related Cplex interface) as programming language. In order to separate cuts for both MILP1' and MILP2', we use the heuristic procedures HP1 and HP2, applied to any pair of paths $p' \in P$, $p \in \Pi_{v,p'}$. The cut separation procedure (implemented using the Cplex featured *user-cuts callback*) works at each branch-and-bound node and iterates until no violated inequalities are found. Concerning the solution of MILP2', at each branch-and-bound node we apply both the constraint separation procedure A and the cut separation procedure. In particular, constraints (29) are separated only if the current linear relaxation is integer (using the Cplex featured *lazy-constraints callback*), since computational results show a remarkable increase in the running times if we look for violated constraints also at fractional solutions. By combining different models and cut separation procedures, we have obtained and tested several algorithm configurations for DMP. In the following, we report on the ones with better performance, which are described in Table 1: each column is devoted to a configuration and shows its name in the first row and, in the following rows, the model solved by the configuration, the constraint defining the feasible region and the related separation procedure (for models MILP2 and MILP2'), and, in the last row, if cuts (34) are added to strengthen the formulation together with, in this case, the used separation procedure (HP1 or HP2).

The test benchmark is made of two groups: Group I is made of 236 instances derived from different subnetworks of the intermodal public transportation network around Bruxelles (Heilporn et al. 2008); Group II includes 64 instances inspired by the railway system in the Randstad, in the Netherlands (Dollevoet et al. 2015). Notice that we needed to adapt the latter set of instances as follows. Data from (Dollevoet et al. 2015) are related to an event-activity network of type *Dep-Arr*, so we shrink it to an *Arr-Arr graph* (see Heilporn et al. 2008). Moreover, the data describe the path and times of each single vehicle and vehicles on different lines run according to different time periods: we consider instead a fixed time period T coherent with the first group of instances, and we consider the lines along which vehicles move. The number of passengers provided is related to their origin and destination stations and the time in which people start their journey, without information on which path is used in between, while we need passengers associated to each route,

Table 2 Instance summary: size groups.

Group		min #var.	max #var.	min #constr.	max #constr.	#instances	T period
I	S	136	336	99	246	184	30
	L	382	12833	309	19023	52	30
II	L	610	4522	508	7762	64	30

Table 3 Linear relaxation without cuts.

Group I	D	MILP1		MILP2		MILP1'		MILP2'			
		%int	Gap %	time	%solved	time	%int	Gap%	time	%solved	time
I S	all	50.00	5.14	0.05	100.00	0.55	72.28	4.64	0.05	100.00	0.41
	4	91.30	6.31	0.05	100.00	0.10	95.65	1.71	0.05	100.00	0.06
	8	41.30	6.23	0.05	100.00	0.44	67.39	5.76	0.05	100.00	0.28
	18	32.61	5.84	0.04	100.00	0.91	56.52	4.74	0.05	100.00	0.67
	28	34.78	3.30	0.05	100.00	0.73	69.57	3.69	0.04	100.00	0.65
I L	all	69.23	6.62	0.23	92.31	3.14	78.85	6.02	0.30	94.23	8.05
	4	100.00	0.00	0.22	92.31	1.32	100.00	0.00	0.27	100.00	21.73
	8	69.23	3.33	0.22	92.31	2.26	76.92	2.01	0.31	92.31	1.36
	18	53.85	7.32	0.23	92.31	4.16	69.23	5.66	0.31	92.31	4.05
	28	53.85	8.12	0.24	92.31	4.82	69.23	9.39	0.31	92.31	3.93
average		54.24	5.36	0.09	98.31	1.08	73.73	4.88	0.10	98.73	2.02

thus we aggregate the number of passengers from each origin to each destination within the provided time window and re-distribute them on the routes we take into account. In particular, we create instances in which passengers use routes with at most one, two or three connection arcs (with no restriction in the total number of arcs) or at most a total of five arcs (with no restriction in the number of connections). Different instances are obtained considering a different network or a different initially delayed node. As from Table 2, instances of Group I have been divided into two size classes S and L, according to the number of variables and constraints needed by model MILP1: S includes the instances where the number of variables times the number of constraints is not larger than 100000, L the remaining instances. Group II only has instances in class L. The initial delay D is equal to 4, 8, 18 or 28 minutes and the period T between two vehicles of the same line has been set to 30 minutes. In the following, we report the results obtained on a 2 GHz Intel Core i7 processor, 64 bit operating system with 4GB RAM, with Cplex presolve, heuristics and general purpose cuts switched off, and a time limit of 300 seconds per instance.

A first set of experiments performed on Group I instances aims at verifying the impact of the proposed model and cuts improvements on the strength of the formulations for DMP, in terms of bound provided by the linear relaxation. Results are reported in Tables 3 and 4. Rows aggregate the instances by size class (S or L) and by initial delay D (“all” refers to the average of all instances within the same size class), and, further, the last row reports average results for all the instances.

Table 4 Linear relaxation with cuts (MILP1'+HP1 and +HP2).

Group I	D	#inst2	MILP1'+HP1					MILP1'+HP2				
			%solved2	%int2	Gap2%	#cuts	time2	%solved2	%int2	Gap2%	#cuts	time2
I S	all	51	100.00	43.14	6.51	69.76	0.49	100.00	74.51	3.05	80.18	0.27
	4	2	100.00	0.00	1.71	0.00	0.12	100.00	0.00	0.99	1.00	0.12
	8	15	100.00	46.67	7.23	43.00	0.35	100.00	80.00	5.26	47.07	0.16
	18	20	100.00	35.00	5.37	88.65	0.56	100.00	75.00	2.00	104.10	0.34
	28	14	100.00	57.14	10.89	81.43	0.58	100.00	78.57	3.97	92.79	0.32
I L	all	11	72.73	50.00	6.24	85.13	1.34	81.82	77.78	1.92	697.33	33.68
	4	0	-	-	-	-	-	-	-	-	-	-
	8	3	66.67	100.00	0.00	29.00	0.62	100.00	100.00	0.00	1790.33	99.24
	18	4	75.00	33.33	1.19	74.00	0.94	75.00	66.67	0.90	73.67	0.48
	28	4	75.00	33.33	11.29	133.67	2.24	75.00	66.67	2.94	228.00	1.31
average			95.16	44.07	6.47	71.85	0.60	96.77	75.00	2.90	172.75	5.28

-: no instance is taken into account in this group.

The columns of Table 3 are divided into two parts, related respectively to models MILP1 and MILP2, and to MILP1' and MILP2'. It presents results on the effect of using tighter constants in the new models. In column *%int* the number of instances that are solved to integrality by the linear relaxation are reported, while column *Gap%* shows the average percent integrality gap for the remaining instances, that is the difference between the optimal integer value and the value of the linear relaxation, divided by the optimal integer value and multiplied by 100. We remark that, since MILP1 and MILP2 have the same linear relaxation, they share columns *%int* and *Gap%*. The same holds for MILP1' and MILP2'. As we can easily see, the percentage of instances solved to integrality increases thanks to the stronger constants, and the overall Gap for the remaining instances is reduced. In general, we observe that the improvements to the model coefficients are quite effective: the number of instances that are solved to integrality by the linear relaxation of MILP1' and MILP2' is always remarkably larger (from 54.24% to 73.73% on average), and even the gap for the remaining instances is slightly reduced (from 5.36% to 4.88%). The third column of each part of Table 3 reports the running times of MILP1 and MILP1' models, whose linear relaxations have been always solved within the time limit. In the last two columns of each part, we show results for MILP2 and MILP2', namely the percentage of instance whose linear relaxation is solved within the time limit and the related computing time (average on solved instances). We notice that MILP1 and MILP1' not only solve all the instances within the time limit, but the running time is always fairly smaller than MILP2 and MILP2'. As a matter of fact, the proposed constraint separation procedure shows rather slow, so that, in the following, we will focus on results from MILP1 and MILP1' models.

Table 4 shows the impact of the new cuts, generated using the two different heuristic approaches HP1 and HP2. To this end, from now on, we just consider those instances such that the linear

Table 5 Branch-and-bound: results for MILP1' and MILP1'+HP2.

Group	D	MILP1'		MILP1'+HP2				Cut-and-branch MILP1'+HP2				
		#nodes	time2	%solved2	#nodes	#cuts	time2	%solved2	#nodes	#cuts	time2	%fast
I S	all	8.29	0.14	100.00	2.10	80.49	0.33	100.00	3.98	10.01	0.11	62.75
	4	3.00	0.12	100.00	3.00	1.00	0.27	100.00	1.00	2.00	0.09	100.00
	8	4.40	0.10	100.00	1.87	46.87	0.23	100.00	3.40	11.00	0.11	26.66
	18	9.75	0.14	100.00	1.75	104.15	0.37	100.00	3.95	10.10	0.11	70.00
	28	11.14	0.16	100.00	2.71	94.07	0.39	100.00	5.07	10.21	0.13	85.71
I L	all	123.27	2.59	81.82	2.00	703.56	33.97	100.00	218.00	2.54	2.54	36.36
	4	-	-	-	-	-	-	-	-	-	-	-
	8	4.00	0.39	100.00	1.00	1790.33	99.75	100.00	2.33	11.00	0.37	0.00
	18	140.75	3.14	75.00	1.67	73.67	0.49	100.00	117.75	10.50	1.88	50.00
	28	195.25	3.70	75.00	3.33	246.67	1.66	100.00	480.00	11.00	4.83	50.00
average		28.69	0.57	96.77	2.10	86.61	0.40	100.00	41.95	10.21	0.54	58.06
II L	all	2.31	0.36	34.38	1.00	0.00	2.73	50.00	1.75	3.19	45.95	0.00
	4	-	-	-	-	-	-	-	-	-	-	-
	8	-	-	-	-	-	-	-	-	-	-	-
	18	2.81	0.40	50.00	1.00	0.00	2.75	56.25	1.22	1.22	16.07	0.00
	28	4.44	0.42	18.75	1.00	0.00	2.67	43.75	2.43	5.71	84.38	0.00

-: no instance is taken into account in this group.

relaxation of MILP1' and MILP2' does not provide the integer optimal solution without applying cuts (the same instances considered for the column *Gap%* of the second half of Table 3): column *#inst2* shows their number. With reference to these *second round* instances, the column *%solved2* reports the percentage of instances solved within the time limit. Columns *%int2* and *Gap2%* show respectively the percentage of instances solved to integrality and the percent integrality gap for the remaining ones. Column *#cuts* shows the number of cuts separated and *time2* is the computing time. Results show that separating inequalities (34) yields further sensible improvements, both in terms of percentage of instances closed at the root node, and integrality gap in the remaining cases.

It can be seen how procedure HP2 has better performance on MILP1' model, as it solves the same number of instances as HP1 in smaller running times for all instance subgroups but I L with initial delay of 8 minutes: in this case, we are solving one extra instance that needs a large number of cuts and more running time than the average to be solved. For this reason, test on the branch-and-cut procedure considers the algorithm configuration MILP1'+HP2.

The branch-and-cut procedure has been tested on instances of both Group I and Group II and results are presented in Table 5, where we consider only the second round instances, that is the ones that are not solved to integrality at the root node without adding any cut: they correspond to instances of Table 4 and to all the instances of Group II with delay 18 and 28 minutes. The 3rd and 4th columns of this table show results regarding the pure branch-and-bound procedure, while the next 4 columns show how the new cuts (configuration MILP1'+HP2) impact on the solution

process in terms of number of instances solved within the time limit and, with reference to these instances, average number of branch-and-bound nodes, average number of applied cuts and average running times. We remark that the row “average” refers to the instances of Group I. In general, the number of nodes in the branching tree is reduced, due to the addition of cuts, which confirms the effectiveness of the proposed cuts. However, the separation procedures seem to lack in efficiency, so that the running times are definitively larger than the ones needed by MILP1’. In fact, HP2 enumerates all possible pairs of paths $p' \in P$, $p \in \Pi_{v_{p'}}$, and their number may be prohibitively large, especially for the larger instances. In particular, as soon as the cut separation procedure is triggered and starts providing violated inequalities, no instance of Group II can be solved, so that just instances with integer linear relaxation for MILP1’ (among which all the ones with smaller initial delays) terminate within the time limit.

The running times of the branch-and-cut procedures may be reduced by generating a small subset of cuts, in order to limit the computational effort devoted to separation. To this end, we performed some preliminary experiments in which we selected just a few cuts to be added at the root node. We recall that HP2 derives a cut (34) from a violated inequality (32) and we observe that a stronger cut may be obtained by choosing the paths p'' in (32) equal to p' whenever possible. Thus, in the attempt of selecting more effective cuts, we modify HP2 by choosing

$$p''_{\alpha(p,i,j)} = \begin{cases} p' & \text{if } p' \in A_p^C \\ \arg \max_{p \in P:(i,j) \in A_p^C} z_p^* & \text{otherwise} \end{cases}$$

where z_p^* is the current fractional solution. The last five columns of Table 5 report the results of this cut-and-branch method where, after a preliminary tuning on the number of cuts to be added to the root node, a maximum of 11 cuts are added at the root node. We report, for each group, the average number of cuts, the average number of processed nodes, the average running time and, in the last column, the percentage of instances where the cut-and-branch procedure is faster than MILP1’. In general, the cut-and-branch approach solves a larger percentage of instances within the time limit. Moreover, as expected, the average number of nodes is smaller than MILP1’ for every group but for instances of type I-L with 28 minutes of initial delay: this is due to a single instance showing an unusual large number of nodes (731 for MILP1’ and 1887 for cut-and-branch, which also affect the average computing time in this group). In fact, about 58% of the instances of Group I are solved faster, in particular the small ones, as well as instances with small or close to the period T initial delay. Within the same group, the cut-and-branch procedure seems to be faster when the maximum number of connection arcs per passenger path is small (typically 1): in this case, added cuts involve less variables and seem to be more effective.

Concerning instances of Group II, adding a limited number of cuts yields more instances solved within the time limit than MILP1'+HP2, at the cost of larger running times, mostly spent by the separation procedure, so that MILP1' alone is always faster on Group II instances.

The overall results for instances of Groups I and II show that, in order to combine the advantages of a tightened formulation together with reduced running times, a way to select, among all the violated cuts, the most effective ones, as well as separation procedures able to more efficiently deal with large number of paths should be investigated.

6. Conclusions

In this paper we have presented strengthened formulations for the Delay Management Problem, starting from two formulations MILP1 and MILP2 taken from literature. A first improvement has been obtained by tightening the bounds for the model coefficients, leading to a new formulation MILP1'. By projection, we have obtained a further new formulation MILP2': thanks to the selection of restricted sets of paths, the number of constraints is sensibly reduced with respect to MILP2, even if it remains exponential in the worst case.

We have also proposed new classes of valid inequalities for the new formulations, obtained by studying one single MILP2' constraint and deriving facets for the related polytope: we have shown that the new inequalities generalize the ones presented in literature and provide stronger cuts.

In order to analyze the potential of these theoretical results in computational terms, we have implemented different branch-and-cut algorithms for DMP. To this end, preliminary separation procedures have been provided for both MILP2' constraints and the new family of cuts. Results on a benchmark of instances derived from real transportation networks show that the use of the new formulations and cuts can lead to more efficient solution methods.

In fact, the performance of the models with improved constants are sensibly better, in terms of instances solved at the root node, integrality gap, number of branch-and-bound nodes and efficiency. In particular MILP1' is sensibly faster than MILP1 and MILP2. Concerning MILP2', a better implementation may be obtained by investigating more efficient constraint separation procedures that do not enumerate all passenger paths.

Also, computational results show that the cuts are quite effective in tightening the linear relaxation and reducing the number of branch-and-cut nodes. Even in this case, the cut separation procedures should be fastened, by investigating cut selection procedures that avoid the enumeration of all possible pairs of paths. In this direction, some preliminary experiments, where a restricted number of selected cuts is applied, often provide sensible improvements in the overall efficiency and confirm the potential of the proposed family of cuts.

The investigation of more efficient separation procedures for MILP2' constraints and for the new improved cuts as well as their integration into branch-and-cut methods for DMP are the object of ongoing research.

Further research could also consider to extend these models to deal with more than one initial delay: model MILP1' is easily extended, as it suffices to add a suitable constraint for each distinct initial delay. An extension of model MILP2' has to take into account the propagation of the different delays, so that it requires different families of extra constraints and related cuts should be studied.

Appendix A: Proof of Proposition 2

PROPOSITION 2. *The following formulation MILP2' is valid for DMP and the value of the linear relaxation is the same as MILP1':*

$$\begin{aligned}
 (MILP2') \quad & \min \sum_{p \in P} c_p [(1 - z_p)T + u_p] \\
 \text{s.t.} \quad & u_{p'} \geq D - \sum_{(i,j) \in A_p^R} \sigma_{ij} - \sum_{(i,j) \in A_p^C} \left[\sigma_{ij} z_{p''_{\alpha(p,i,j)}} + M_i (1 - z_{p''_{\alpha(p,i,j)}}) \right] - M_{v_{p'}} (1 - z_{p'}) \\
 & \forall p' \in P, p \in \Pi_{v_{p'}}, (p''_1, p''_2 \dots p''_{|A_p^C|}) : p''_1 \in P_1^p, p''_2 \in P_2^p \dots p''_{|A_p^C|} \in P_{|A_p^C|}^p
 \end{aligned} \tag{29}$$

$$u_p \geq 0 \quad \forall p \in P \tag{30}$$

$$z_p \in \{0, 1\} \quad \forall p \in P. \tag{31}$$

Proof. The result is obtained by showing that the feasible domain of MILP2' is the projection of the feasible domain of MILP1' onto the space of u and z variables. This can be done by applying Fourier-Motzkin elimination to d variables with steps similar to the ones used by Heilporn et al. (2008) to obtain MILP2 from MILP1. First, we rewrite the constraints (21) and (22): from (22) we have

$$d_j \geq d_i - \min_{p \in P: (i,j) \in A_p^C} \{ \sigma_{ij} z_p + M_i (1 - z_p) \} \quad \text{if } (i, j) \in A_p^C$$

and, by defining for each arc (i, j)

$$l_{ij} = \begin{cases} \min_{p \in P: (i,j) \in A_p^C} \{ \sigma_{ij} z_p + M_i (1 - z_p) \} & \text{if } (i, j) \in A_p^C \\ \sigma_{ij} & \text{if } (i, j) \in A^R, \end{cases}$$

we can write

$$d_j \geq d_i - l_{ij} \quad \forall (i, j) \in A^R \cup A_p^C.$$

Now, let p be one of the paths starting at the delayed node 1. For ease of notation, we denote with 1 the first node in path p , with 2 the second and so on, until the last node v_p . By constraints (20), (21) and (22)

$$d_1 \geq D, \quad d_2 \geq d_1 - l_{12} \geq D - l_{12}, \quad d_3 \geq d_2 - l_{23} \geq D - l_{12} - l_{23}$$

...

$$d_{v_p} \geq D - \sum_{(i,j) \in A_p} l_{ij}.$$

Similarly, for any path p whose first node is $k \neq 1$, starting from $d_k \geq 0$, we have

$$d_{v_p} \geq 0 - \sum_{(i,j) \in A_p} l_{ij}.$$

Hence, we have

$$\begin{aligned} d_{v_p} &\geq D - \sum_{(i,j) \in A_p} l_{ij} && \forall p \in \bar{\Pi} : a_p = 1 \wedge A_p^C \subseteq A_P^C \\ d_{v_p} &\geq 0 - \sum_{(i,j) \in A_p} l_{ij} && \forall p \in \bar{\Pi} : a_p \neq 1 \wedge A_p^C \subseteq A_P^C \end{aligned}$$

Notice that, since l_{ij} is nonnegative, the last inequalities are redundant and can be omitted.

We remark also that paths $p \in \bar{\Pi}$ with at least one connection not included in any passenger path are not considered, since the delay does not spread on them, according to constraints (22).

We recall that, by (23),

$$u_p + M_{v_p}(1 - z_p) \geq d_{v_p}, \quad \forall p \in P.$$

As it holds for every $p \in P$, it also holds for a $p' \in P$ such that $v_{p'} = v_p$:

$$u_{p'} + M_{v_{p'}}(1 - z_{p'}) \geq d_{v_{p'}}$$

Noticing that

$$d_{v_{p'}} \geq D - \sum_{(i,j) \in A_p} l_{ij} \quad \forall p \in \bar{\Pi}, a_p = 1 \wedge v_p = v_{p'} \wedge A_p^C \subseteq A_P^C$$

and recalling that just variables z and u appear in the objective function, we can apply Fourier-Motzkin elimination and replace constraints (20), (21), (22) and (23) by

$$\begin{aligned} u_{p'} &\geq D - \sum_{(i,j) \in A_p} l_{ij} - M_{v_{p'}}(1 - z_{p'}) \\ &\forall p' \in P, p \in \bar{\Pi} : a_p = 1 \wedge v_p = v_{p'} \wedge A_p^C \subseteq A_P^C. \end{aligned} \quad (40)$$

Let us now consider a passenger path $p' \in P$ such that $\hat{p}_{v_{p'}}$, i.e. the path from the delayed node 1 to $v_{p'}$ with no connection arcs, exists. Let us define, for ease of notation, $\bar{p} = \hat{p}_{v_{p'}}$. In this case

$$D - \sum_{(i,j) \in A_{\bar{p}}} l_{ij} = D - \sum_{(i,j) \in A_{\bar{p}}} \sigma_{ij} = D - \Theta_{v_{p'}}$$

and one of the constraints (40) is

$$u_{p'} \geq D - \Theta_{v_{p'}} - M_{v_{p'}}(1 - z_{p'}).$$

Consider also another path $p \in \bar{\Pi}$ from 1 to $v_{p'} = v_p$ such that

$$\sum_{(i,j) \in A_p^R} \sigma_{ij} \geq \Theta_{v_{p'}}.$$

The resulting constraints of type (40) is

$$u_{p'} \geq D - \sum_{(i,j) \in A_p} l_{ij} - M_{v_{p'}}(1 - z_{p'}) \quad (41)$$

which is redundant. In fact, as $\sigma_{ij} \leq M_i$

$$\begin{aligned} \sum_{(i,j) \in A_p} l_{ij} &= \sum_{(i,j) \in A_p^R} \sigma_{ij} + \sum_{(i,j) \in A_p^C} \sigma_{ij} + \\ &\quad + \sum_{(i,j) \in A_p^C} \min_{q: (i,j) \in A_q^C} \{(M_i - \sigma_{ij})(1 - z_q)\} \geq \\ &\geq \sum_{(i,j) \in A_p} \sigma_{ij} \geq \Theta_{v_{p'}}. \end{aligned}$$

Summarizing, $\sum_{(i,j) \in A_p} l_{ij} \geq \Theta_{v_{p'}}$, and hence

$$D - \sum_{(i,j) \in A_p} l_{ij} - M_{v_{p'}}(1 - z_{p'}) \leq D - \Theta_{v_{p'}} - M_{v_{p'}}(1 - z_{p'})$$

meaning that (41) is redundant.

The discussion above shows that MILP1' is equivalent to

$$\begin{aligned} \min \quad & \sum_{p \in A} c_p [(1 - z_p)T + u_p] \\ \text{s.t.} \quad & u_{p'} \geq D - \sum_{(i,j) \in A_p} l_{ij} - M_{v_{p'}}(1 - z_{p'}) \quad \forall p' \in P, p \in \Pi_{v_{p'}} \quad (42) \\ & u_p \geq 0 \quad \forall p \in P \\ & z_p \in \{0, 1\} \quad \forall p \in P \end{aligned}$$

In case $(i, j) \in A^C$, the term l_{ij} is not linear and can be written as follows:

$$\begin{aligned} l_{ij} &= \min_{p \in P: (i,j) \in A_p^C} \{\sigma_{ij} z_p + M_i(1 - z_p)\} = \\ &= \min_{p \in P: (i,j) \in A_p^C} \{M_i - (M_i - \sigma_{ij})z_p\} = \\ &= M_i - \max_{p \in P: (i,j) \in A_p^C} \{(M_i - \sigma_{ij})z_p\} = \\ &= M_i - (M_i - \sigma_{ij})\tilde{z}_{ij} \end{aligned} \quad (43)$$

where $\tilde{z}_{ij} = \max_{p \in P: (i,j) \in A_p^C} z_p$.

Then MILP1' is equivalent to

$$\text{(MILP2)*} \quad \min \quad \sum_{p \in P} c_p [(1 - z_p)T + u_p] \quad (44)$$

$$\text{s.t.} \quad u_{p'} \geq D - \sum_{(i,j) \in A_p^R} \sigma_{ij} - \sum_{(i,j) \in A_p^C} M_i + \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij})\tilde{z}_{ij} - M_{v_{p'}}(1 - z_{p'}) \quad \forall p' \in P, p \in \Pi_{v_{p'}} \quad (45)$$

$$u_p \geq 0 \quad \forall p \in P \quad (46)$$

$$z_p \in \{0, 1\} \quad \forall p \in P \quad (47)$$

This model is still not linear, because of the term \tilde{z}_{ij} . In order to get a linear formulation, we write the same inequality involving \tilde{z} for each combination of paths in P such that the t -th path in the combination includes the t -th connection arc of p . The resulting constraints are

$$\begin{aligned} u_{p'} \geq D - \sum_{(i,j) \in A_p^R} \sigma_{ij} - \sum_{(i,j) \in A_p^C} M_i + \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij})z_{p''_{\alpha(p,i,j)}} - M_{v_{p'}}(1 - z_{p'}), \\ \forall p' \in P, p \in \Pi_{v_{p'}}, (p''_1, p''_2 \dots p''_{|A_p^C|}) : p''_1 \in P_1^p, p''_2 \in P_2^p \dots p''_{|A_p^C|} \in P_{|A_p^C|}^p \end{aligned}$$

yielding formulation MILP2'. □

Appendix B: Proofs of technical lemmas

LEMMA 1. For any path $p \in \bar{\Pi}$ such that $a_p = 1$, $M_{v_p} \geq D - \sum_{(i,j) \in A_p} \sigma_{ij}$.

Proof. Let k be one of the nodes of p and denote by $p(k)$ the sub-path of p from 1 to k . By defining

$$\tilde{M}_k = \max\{0, D - \sum_{(i,j) \in A_{p(k)}} \text{slack}_{ij}\}$$

and

$$\tilde{\sigma}_{ij} = \min\{\text{slack}_{ij}, \tilde{M}_i\}.$$

we have, for each node k included in the path p , $\tilde{M}_k = D - \sum_{(i,j) \in A_{p(k)}} \tilde{\sigma}_{ij}$. By the definition of M_k , we have $\tilde{M}_k \leq M_k$ and, by definition of σ_{ij} we have

$$\tilde{\sigma}_{ij} \leq \sigma_{ij}, \forall (i,j) \in A_p$$

and hence

$$M_{v_p} \geq \tilde{M}_{v_p} = D - \sum_{(i,j) \in A_p} \tilde{\sigma}_{ij} \geq D - \sum_{(i,j) \in A_p} \sigma_{ij}.$$

□

LEMMA 2. Assuming that $D > \sum_{(i,j) \in A_p} \sigma_{ij}$, $W \cup \{p'\}$ is a minimal cover of $Y^{fDMP'}$ and the inequality

$$z_{p'} + \sum_{w \in W} z_w \leq |W| \quad (48)$$

defines a facet of $\text{conv}(Y^{fDMP'}|_{u_{p'}=0})$.

Proof. First, we prove that the set is a cover. In fact, as by hypothesis $D > \sum_{(i,j) \in A_p} \sigma_{ij}$,

$$\begin{aligned} \sum_{w \in W \cup \{p'\}} \gamma_w &= M_{v_{p'}} + \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) = M_{v_{p'}} + \sum_{(i,j) \in A_p^C} M_i - \sum_{(i,j) \in A_p} \sigma_{ij} + \sum_{(i,j) \in A_p^R} \sigma_{ij} > \\ &> \sum_{(i,j) \in A_p^R} \sigma_{ij} + \sum_{(i,j) \in A_p^C} M_i + M_{v_{p'}} - D. \end{aligned}$$

In order to prove that the cover is minimal, we show that $W \cup \{p'\} \setminus \bar{w}$ is not a cover for each $\bar{w} \in W \cup \{p'\}$.

We have two cases: $\bar{w} = p'$ and $\bar{w} \in W$.

If $\bar{w} = p'$, recalling that $A_p^C(w)$ denotes the set $\{(h,l) \in A_p^C | p''_{\alpha(p,h,l)} = w\}$,

$$\begin{aligned} \sum_{w \in W \cup \{p'\} \setminus \bar{w}} \gamma_w &= \sum_{(i,j) \in A_p^C \setminus A_p^C(p')} \gamma_w \leq \sum_{w \in W} \gamma_w = \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) = \\ &= \sum_{(i,j) \in A_p^C} M_i - \sum_{(i,j) \in A_p} \sigma_{ij} + \sum_{(i,j) \in A_p^R} \sigma_{ij} \leq \\ &\leq \sum_{(i,j) \in A_p^R} \sigma_{ij} + \sum_{(i,j) \in A_p^C} M_i + M_{v_{p'}} - D \end{aligned}$$

as $M_{v_p} = M_{v_{p'}}$ and, by Lemma 1, $M_{v_p} \geq D - \sum_{(i,j) \in A_p} \sigma_{ij}$.

If $\bar{w} \in W$, we have

$$\begin{aligned} \sum_{w \in W \cup \{p'\} \setminus \bar{w}} \gamma_w &= M_{v_{p'}} + \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) - \sum_{(h,l) \in A_p^C(\bar{w})} (M_h - \sigma_{hl}) = \\ &= M_{v_{p'}} + \sum_{(i,j) \in A_p^C} M_i - \sum_{(i,j) \in A_p} \sigma_{ij} + \sum_{(i,j) \in A_p^R} \sigma_{ij} - \sum_{(h,l) \in A_p^C(\bar{w})} M_h + \sum_{(h,l) \in A_p^C(\bar{w})} \sigma_{hl}. \end{aligned}$$

We have to show that

$$\sum_{w \in W \cup \{p'\} \setminus \bar{w}} \gamma_w \leq \sum_{(i,j) \in A_p^R} \sigma_{ij} + \sum_{(i,j) \in A_p^C} M_i + M_{v_{p'}} - D,$$

that is,

$$\sum_{(h,l) \in A_p^C(\bar{w})} M_h \geq D - \sum_{(i,j) \in A_p} \sigma_{ij} + \sum_{(h,l) \in A_p^C(\bar{w})} \sigma_{hl}.$$

Now, let (h_1, j) be the first connection arc of $A_p^C(\bar{w})$ found on path p moving from node 1. Denoting with $p(h_1)$ the sub-path of p from 1 to h_1 , we have

$$\begin{aligned} \sum_{(h,l) \in A_p^C(\bar{w})} M_h &\geq M_{h_1} \geq \text{(by Lemma 1)} \\ &\geq D - \sum_{(i,j) \in A_p(h_1)} \sigma_{ij} = D - \left(\sum_{(i,j) \in A_p} \sigma_{ij} - \sum_{(i,j) \in A_p^C(\bar{w})} \sigma_{ij} - \sum_{(i,j) \in (A_p \setminus A_p(h_1)) \setminus A_p^C(\bar{w})} \sigma_{ij} \right) \geq \\ &\geq D - \sum_{(i,j) \in A_p} \sigma_{ij} + \sum_{(i,j) \in A_p^C(\bar{w})} \sigma_{ij}. \end{aligned}$$

□

LEMMA 3 If $W \neq \emptyset$, for any $w \in W \cup \{p'\}$, $\gamma_w \leq \sum_{(i,j) \in A_p^R} \sigma_{ij} + \sum_{(i,j) \in A_p^C} M_i + M_{v_{p'}} - D$.

Proof. We write

$$\begin{aligned} RHS &= \sum_{(i,j) \in A_p^R} \sigma_{ij} + \sum_{(i,j) \in A_p^C} M_i + M_{v_{p'}} - D = \\ &= \sum_{(i,j) \in A_p} \sigma_{ij} - \sum_{(i,j) \in A_p^C} \sigma_{ij} + \sum_{(i,j) \in A_p^C} M_i + M_{v_{p'}} - D = \\ &= \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) + M_{v_{p'}} - \left(D - \sum_{(i,j) \in A_p} \sigma_{ij} \right). \end{aligned}$$

If $w \in W$, we have

$$RHS - \gamma_w = M_{v_{p'}} + \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) - \sum_{(i,j) \in A_p^C(w)} (M_i - \sigma_{ij}) - \left(D - \sum_{(i,j) \in A_p} \sigma_{ij} \right) \geq 0$$

since $A_p^C(w) \subseteq A_p^C$ and $M_{v_{p'}} = M_{v_p} \geq D - \sum_{(i,j) \in A_p} \sigma_{ij}$ by Lemma 1.

If $w = p'$,

$$\begin{aligned} RHS - \gamma_{p'} &= \sum_{(i,j) \in A_p^C} (M_i - \sigma_{ij}) - \sum_{(i,j) \in A_p^C(p')} (M_i - \sigma_{ij}) - \left(D - \sum_{(i,j) \in A_p} \sigma_{ij} \right) = \\ &= \sum_{A_p^C \setminus A_p^C(p')} M_i + \sum_{A_p^C(p')} \sigma_{ij} + \sum_{A_p^R} \sigma_{ij} - D. \end{aligned} \tag{49}$$

Since $W \neq \emptyset$, the set $A_p^C \setminus A_p^C(p') \neq \emptyset$, and let (k, j) be the first connection arc in this set on the way from 1 to $v_{p'}$. Recalling that $p(k)$ is the sub-path of p limited to node k , for M_k , by Lemma 1, we have

$$\begin{aligned} M_k &\geq D - \sum_{A_p(k)} \sigma_{ij} = D - \sum_{A_p^R(k)} \sigma_{ij} - \sum_{A_p^C(k)} \sigma_{ij} \geq \\ &\geq D - \sum_{A_p^R(k)} \sigma_{ij} - \sum_{A_p^C(p')} \sigma_{ij} \end{aligned}$$

since $A_{p(k)}^C \subseteq A_p^C(p')$. Hence, from (49),

$$\begin{aligned} RHS - \gamma_{p'} &\geq M_k + \sum_{A_p^C(p')} \sigma_{ij} + \sum_{A_p^R} \sigma_{ij} - D \geq \\ &\geq D - \sum_{A_{p(k)}^R} \sigma_{ij} - \sum_{A_p^C(p')} \sigma_{ij} + \sum_{A_p^C(p')} \sigma_{ij} + \sum_{A_p^R} \sigma_{ij} - D = \\ &= - \sum_{A_{p(k)}^R} \sigma_{ij} + \sum_{A_p^R} \sigma_{ij} \geq 0 \end{aligned}$$

since $A_{p(k)}^R \subseteq A_p^R$. Summarizing $RHS - \gamma_w \geq 0$, for any $w \in W \cup \{p'\}$. \square

References

- Adenzo-Diaz, B., M. Oliva-Gonzalez, P. Gonzalez-Torre. 1999. On-line timetable re-scheduling in regional train services. *Transportation Research Part B* **33**(6) 387–398.
- Berger, A., R. Hoffmann, U. Lorenz, S. Stiller. 2011. Online railway delay management: hardness, simulation and computation. *Simulation* **87**(7) 616–629.
- Chvátal, V. 1979. A greedy heuristic for the set-covering problem. *Mathematics of operations research* **4**(3) 233–235.
- Corman, F., A. D'Ariano, A.D. Marra, D. Pacciarelli, M. Samà. 2017. Integrating train scheduling and delay management in real-time railway traffic control. *Transportation Research Part E* **105** 213–239.
- De Giovanni, L., G. Heilporn, M. Labbé. 2014. A polyhedral study for delay management in public transportation. *Procedia - Social and Behavioral Sciences* **108** 15–25.
- de Vries, R., B. De Schutter, B. De Moor. 1998. On the max-algebraic models for transportation networks. *Proceedings of the 4th International Workshop on Discrete Event Systems 1998, Cagliari (Italy)* 457–462.
- Dolvoet, T., D. Huisman, L. Kroon, M. Schmidt, A. Schöbel. 2015. Delay management including capacities of stations. *Transportation Science* **49**(2) 185–203.
- Dolvoet, T., D. Huisman, M. Schmidt, A. Schöbel. 2010. Delay management with re-routing of passengers. *Transportation Science* **46**(1) 74–89.
- Elmaghraby, S. 1977. *Activity Networks*. Wiley Interscience Publication.
- Gatto, M., R. Jacob, L. Peeters, P. Widmayr. 2007. On-line delay management on a single train line. *Algorithmic Methods for Railway Optimization* 306–320.
- Gatto, M., R. Jacob, L. Peeters, A. Schöbel. 2005. The computational complexity of delay management. *31st Workshop in Theoretic Concepts in Computer Science*.
- Ginkel, A., A. Schöbel. 2007. To wait or not to wait? the bicriteria delay management problem in public transportation. *Transportation Science* **41**(4) 527–538.
- Goverde, R. M. P. 1998. C. A. Brebbia, J. Allan, B. Mellitt, G. Sciutto, eds., *Computer in railways VI*, chap. The max-plus algebra approach to railway timetable. WIT press, Southampton (UK).

- Heilporn, G., L. De Giovanni, M. Labbé. 2008. Optimization models for the single delay management problem in public transportation. *European Journal of Operational Research* **189**(3) 762–774.
- IBM. 2014. Cplex optimizer. URL <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>.
- Liebchen, C., M. Schachtebeck, A. Schöbel, S. Stiller, A. Prigge. 2010. Computing delay resistant railway timetables. *Computers & Operations Research* **37** 857–868.
- Marchand, H., L.A. Wolsey. 1997. The 0-1 knapsack problem with a single continuous variable. *Mathematical Programming* **85** 15–33.
- Schachtebeck, M., A. Schöbel. 2010. To wait or not to wait - and who goes first? delay management with priority decisions. *Transportation Science* **44**(3) 307–321.
- Schmidt, M. 2013. Simultaneous optimization of delay management decisions and passenger routes. *Public Transport* **5**(1-2) 125–147.
- Schöbel, A. 2001. A model for the Delay Management Problem based on mixed-integer-programming. *Electronic Notes in Theoretical Computer Science* **50**(1).
- Schöbel, A. 2007. Integer programming approaches for solving the delay management problem. *Lecture Notes in Computer Science* **4359** 145–170.
- Schöbel, A. 2009. Capacity constraints in delay management. *Public Transport* **1**(2) 135–154.
- Schön, C., E. König. 2018. A stochastic dynamic programming approach for delay management of a single train line. *European Journal of Operational Research* **271** 501–518.
- Suhl, L., T. Mellouli. 1999. N.H.M. Wilson, ed., *Computer-Aided Transit Scheduling*, chap. Requirements for, and designs of, an operations control system for railways. Springer, Berlin, 371–390.
- Suhl, L., T. Mellouli, C. Biederbick, J. Goecke. 2001. M. Pursula, J. Niittymäki, eds., *Mathematical Methods on Optimization in Transportation Systems*, chap. Managing and preventing delays in railway traffic by simulation and optimization. Kluwer, Dordrecht.