Optimization models for the single delay management problem in public transportation

Luigi De Giovanni, Géraldine Heilporn, and Martine Labbé

Université Libre de Bruxelles, Computer Science Department

Bd. du Triomphe CP 210/01, 1050 - Bruxelles, Belgium

{gheilpor, ldegiova, mlabbe}@ulb.ac.be

Abstract

Passengers travelling in public transportation networks often have to use different lines to cover the trip from their origin to the desired destination. As a consequence, the reliability of connections between vehicles is a key issue for the attractiveness of the intermodal transportation network and it is strongly affected by some unpredictable events like breakdowns or vehicle delays. In such cases, a decision is required to determine if the connected vehicles should wait for the delayed ones or keep their schedule. The Delay Management Problem (DMP) consists in defining the wait/depart policy which minimizes the total delay on the network. In this work, we present two equivalent Mixed Integer Linear Programming models for the DMP with a single initial delay, able to reduce the number of variables with respect to the formulations proposed by literature. The two models are solved by a

*The research is part of the project “Analysis and Optimization of Intermodal Public Transportation Networks in the Brussels Capital Region” funded by the Brussels-Capital Region in the context of the Prospective Research for Brussels program.
Branch and Cut procedure and by a Constraint Generation approach respectively, and preliminary computational results are presented.

**Keywords:** Combinatorial Optimization, Delay Management, Mixed Integer Linear Programming, Constraint Generation.

# 1 Introduction

The attractiveness of the public transportation network is strongly related to the reliability of intermodal connections. But connections imply passenger transfers from one vehicle to another and can generate important waiting times, in particular when low frequency lines are taken into account. Missing a connection, because of a delayed incoming vehicle, implies waiting for the next one of the same line, thus remarkably increasing the total travel time.

We define an intermodal public transportation network as a set of train, metro, tramway and/or bus lines, the vehicles moving between different stations. Suppose that a vehicle is delayed. The users of this vehicle who want to transfer to another vehicle at a station could miss their connection. In fact, either the other vehicle waits for the delayed one and the transfer is allowed, or it does not wait for the delayed vehicle. If a vehicle waits for the delayed one, the users travelling on it suffer a delay. Also, a delay is caused for passengers wishing to get on this vehicle later on, and possibly for subsequent other vehicles which will have to wait because of this delay. On the other hand, if a vehicle departs on time, only the passengers on the delayed vehicle suffer a delay, but it might be very high, as the passengers should wait for the next vehicle of the missed line.

To avoid passengers missing their transfers, one could force all departing vehicles to wait until the delayed one has arrived. But, in this case, the delay spreads out through the network, thus affecting
many customers. On the other hand, if all vehicles depart on time, no other passengers but the ones on
the delayed vehicle will be concerned: the number of affected passengers is thus minimized but they will
suffer greater delays as they miss their connections. As a consequence, the best decision is generally to
force only a subset of the vehicles to wait for the delayed ones. The delay management problem (DMP)
consists in determining how the other vehicles of the network should react (to wait or not to wait) in
order to minimize the sum of delays of all the passengers at their destinations.

Let \( S \) be the station set, \( V \) be the vehicle set, and suppose vehicle \( i \) arrives at station \( k \) with a delay \( D \). Further, let \( A \) be the set of all possible passenger paths, each path being defined as a sequence of
direct rides between pairs of stations. We will consider a time horizon \( T \) which is the scheduled time
interval between the stops of two vehicles of the same line at a given station. We consider that \( T \) is
identical for all vehicles of the network. Finally, we suppose \( D < T \).

We represent the change from vehicle \( i \in V \) to vehicle \( j \in V \) at station \( l \in S \) by the triplet \((i, j, k)\) and
we call it a connection. Suppose that a delay \( D \) occurs at a station \( k \) for the vehicle \( i \). For the vehicles
\( j \in V \) for which a connection \((i, j, l)\) is possible, \( l \in S \) being any station after station \( k \), there are two
possibilities: either vehicle \( j \) waits for the delayed vehicle \( i \), allowing travellers on vehicle \( i \) to change to
vehicle \( j \) at station \( l \), or \( j \) leaves on time. In the latter case, the users of vehicle \( i \) needing to change to
vehicle \( j \) at station \( l \) have to wait for the next vehicle of the same line (as \( j \)). If vehicle \( j \) waits for vehicle
\( i \), we say the connection is maintained; otherwise it is suppressed. Of course, if vehicle \( j \in V \) waits for vehicle
\( i \) at station \( l \in S \), then it becomes a delayed vehicle too. Thus we also have to decide what the
vehicles \( j' \in V \) should do, where the vehicles \( j' \) are those for which a connection \((j, j', l')\) is possible,
\( l' \in S \) being any station after \( l \) on vehicle \( j' \)’s trip, and so on.

We also consider the possibility of reducing the delays: a slack time is defined for each stop of a vehicle
at a station, for each direct ride of a vehicle from a station to the next one, and for each transfer from
a vehicle to another at a station. For example, if we consider that at least one minute is necessary to change from vehicle $i$ to vehicle $j$ at station $k$, and that the scheduled departure time of vehicle $j$ is three minutes after the scheduled arrival time of vehicle $i$ to station $k$, the slack time is equal to two minutes.

In this paper, we consider the delay management problem from the passenger’s point of view, so that we want to minimize the sum of passenger delays at their destinations.

This problem has been the object of a few papers, mainly by Schoebel. In [15] (to appear as [16]), she presents three equivalent mixed integer models with one criteria. In the first two ones (which are also presented in [14]), she considers exactly the same problem as the one we discuss here. In the third model, she uses Event-Activity Networks [4], i.e. she considers a set of events and a set of activities that link these events. An event is the arrival or the departure of a given vehicle to or from a given station. An activity corresponds to either a direct ride of one vehicle from a given station to the next one, or a stop of a vehicle at a given station, or a change from one vehicle to another one at a given station. She minimizes the sum of delays of passengers at each activity. Furthermore, Schoebel presents in [15] a fourth mixed integer model, which generalizes the previous ones. Indeed, she considers a situation where more than one initial delay may occur, each of them causing some delays on a part of the network.

Schoebel et al. also propose (see [10, 15]) some bicriteria models, minimizing the sum of delays of all vehicles at all stations as well as the number of suppressed connections. Finally, Schoebel presents in [15] a very general bicriteria model, whose goal is to minimize, (i) the number of users who cannot change from one vehicle to another (because of a suppressed connection), and, (ii) the sum of delays that all the other users experience at their destinations. This model generalises all the previous ones. Scholl [17] and Kliewer [12] have also considered this problem, but they minimize the total waiting time of the users. Scholl’s model does not include slack times, thus the initial delay cannot be reduced. Ginkel [9, 10] has presented the same bicriterial model as Schoebel, and has proposed to solve it by Event-Activity
Networks [4].

A problem closely related to the DMP is presented in Adenso-Diaz et al. [1], where the impact of delays on the vehicle schedules (rather than the passenger comfort) is analyzed. An Integer Programming Model is presented where the decision variables establish if a delayed vehicle has to perform the following scheduled task or skip it. The slack times are taken into account, while the changing activities and the missed connections are neglected. A heuristic procedure is proposed, based on backtracking for exploring the solutions space, that reduces the search by means of the elimination of certain branches which are not likely to generate good solutions. The evaluation of the quality of each obtained solution is made on the basis of the priority of each service, the passengers transported and the delays that these passengers have to suffer. The best results are offered to the traffic controller so that, using what-if tools, he may choose the alternative that he considers the most adequate from among these. Among some recent papers concerning the DMP we cite Gatto et al. [7] and [8].

The subject of the DMP was brought up by two large traffic associations serving the states Rheinland-Pfalz and the Saarland (both in Germany). Public transportation companies are interested in analyzing the consequences of delays or changes in the schedule. On a regional train line in Rheinland-Pfalz, the 40km long Lautertalbahn leading from Kaiserslautern to Lauterecken, Deutsche Bahn installed an automatic system informing the bus drivers waiting at the stations about the exact arrival times of the incoming trains. Based on this information, the DMP determines whether the drivers should wait for a delayed train or depart on time, see Schoebel [15]. Another application is under study for the public transportation network in Brussels (Belgium), based on the results presented in this paper. In this case the DMP concerning the connections metro-tram, metro-bus and tram-bus during the off-peak hours is an important issue for the public transport company (STIB) because it is considered as an important factor influencing the quality of service (see De Giovanni [2] and De Giovanni et al. [3]).
In this paper, we propose two new compact models for the DMP. In Section 2, we present a new graph interpretation, from which we derive a first model for the DMP containing three types of variables. We show that this model can be seen as a simplification of Schoebel’s linear mixed integer model presented in [14, 15]. In the third section, we further reduce the number of variables, and obtain a second equivalent model. The first and the second models are solved by a standard MILP solver and by a constraint generation approach respectively. The procedures and some preliminary results are presented in the fourth section. Finally, the fifth section is devoted to some concluding remarks and suggestions for further research on DMP.

2 A new model for the DMP

In this section, we present a new model for the DMP. It exploits a simple network graph representation based on the Event-activity-networks concept (see [4]) and it is very close to the formulation (TDM-B) given by Shoebel [15].

Let $R_i \subseteq S \times S$ be the set of vehicle $i$’s rides from a station to the next one, and $S_i \subseteq S$ the set of stations where vehicle $i$ stops. Furthermore, let $C_a \subseteq V \times V \times S$ be the connection set on path $a \in A$, with $C = \bigcup_{a \in A} C_a$.

We define the “Arr-Arr graph” as a directed graph $G = (N, E)$, where each node corresponds to the arrival of a vehicle at a station. The arcs are divided into two subsets: $arcDA$ and $arcC$. The set $arcDA$ corresponds to direct vehicle rides from a station to the next one (including the stop at the arrival station). The set $arcC$ corresponds to connection rides, defined as a connection between different vehicles at a station followed by the trip of the outgoing vehicle towards the next station. A passenger path $a$ can be seen as an ordinary path on the “Arr-Arr graph”, and $arcC_a$ is the set of corresponding connection
arcs. Thus we have:

- \( N = \{(i, k)_{\text{arr}} : i \in V, k \in S_i \} \subseteq V \times S \),

- \( E = \text{arcDA} \cup \text{arcC} = \{(i, k)_{\text{arr}}, (i, l)_{\text{arr}} : (k, l) \in R_i \}
  \cup \{(i, k)_{\text{arr}}, (j, l)_{\text{arr}} : (i, j, k) \in C \}. \)

We also associate weights \( \text{slack}_{uv} \) to the arcs \((u, v) \in E\), representing the times that can be saved on those arcs. In particular, if \((i, j) \in \text{arcDA}\), \(\text{slack}_{ij}\) includes the time that can be saved for the stop of a vehicle at a station and for the trip between two stations. If \((i, j) \in \text{arcC}\), \(\text{slack}_{ij}\) summarizes the time that can be saved for the passenger transfer plus the one for the trip of the outgoing vehicle towards the next station. Finally, we associate variables \(d_u\) to nodes \(u \in N\), which correspond to the arrival delays at these nodes.

We illustrate this definition of the Arr-Arr graph with the example network depicted in Figure 1. We have two vehicles \(i, j\) and five stations \(h, k, l, m, n\). The nodes of the Arr-Arr graph for this network are \((i, k)_{\text{arr}}, (j, n)_{\text{arr}}, (i, l)_{\text{arr}}\) and \((j, k)_{\text{arr}}\). The direct vehicle rides are \(((i, k)_{\text{arr}}, (i, l)_{\text{arr}})\) and \(((j, k)_{\text{arr}}, (j, n)_{\text{arr}})\), and the connection rides are \(((i, k)_{\text{arr}}, (j, n)_{\text{arr}})\) and \(((j, k)_{\text{arr}}, (i, l)_{\text{arr}})\). The related Arr-Arr graph is represented in Figure 2.

Note that this graph representation can be seen as a simplification of the graph representation used in [9, 14, 15], where additional nodes \((i, k)_{\text{dep}} : i \in V, k \in S_i \) are used to represent the departures of vehicles from stations. Indeed, the Arr-Arr graph can be obtained by a node reduction operation to delete all departure nodes, which is in fact a significant reduction of the problem size.

Let \(A\) be the set of all possible passenger paths, \(p_a\) be the number of passengers on \(a\), and \(v_a \in N\) the last node on path \(a\). Further, we define a variable \(d_i\) which represents the arrival delay at node \(i\), for
Figure 1: An example of public transportation network.

Figure 2: Arr-Arr graph of the network of Figure 1.
each \( i \in V \). Finally, let \( z_a \) be a boolean variable that says if all connections on path \( a \) are maintained or not:

\[
z_a = \begin{cases} 
1 & \text{if all connections on path } a \text{ are maintained,} \\
0 & \text{otherwise,}
\end{cases}
\]

and \( u_a \) be a variable which represents the delay at the last node of path \( a \) if \( z_a = 1 \) and is 0 otherwise:

\[
u_a = \begin{cases} 
v_{va} & \text{if all connections on path } a \text{ are maintained,} \\
0 & \text{otherwise.}
\end{cases}
\]

Suppose that there is an initial delay at node 1. The DMP can be formulated as follows:

\[
(MILP1) \quad \min \sum_{a \in A} p_a [(1 - z_a)T + u_a]
\]

\text{s.t.:}

\[
d_1 = D
\]

\[
d_i - d_j \leq \text{slack}_{ij} \quad \forall (i, j) \in \text{arcDA}
\]

\[
d_i - d_j + M_{ij} z_a \leq \text{slack}_{ij} + M_{ij}
\]

\[
\forall a \in A, (i, j) \in \text{arcC}_a
\]

\[
T z_a - u_a + d_{va} \leq T \quad \forall a \in A
\]

\[
d_i \geq 0 \quad \forall i \in N
\]

\[
u_a \geq 0 \quad \forall a \in A
\]

\[
z_a \in \{0, 1\} \quad \forall a \in A,
\]

where \( M_{ij} = \text{slack}_{ij} + T \).
The objective is to minimize the total sum of passenger delays, once they have arrived at their destinations. Note that a delay of $T$ (corresponding to the common scheduled time interval between vehicles) is considered for passengers missing at least one connection on their paths (they have to wait for the next vehicle). Constraint (1) gives the initial delay. Constraint (2) says that the delay at the end of a direct ride must be greater or equal to the delay at the beginning of this ride, minus perhaps the time that can be saved thanks to the slack on this ride. Constraint (3) says that, if all connections are maintained on path $a$, we must have the same constraint on the delays for connection rides as the one for the direct rides. If at least one connection is not maintained on path $a$, this constraint is redundant (thanks to constant $M_{ij}$ definition and the assumption $D < T$). Constraint (4) says that if all connections on path $a$ are maintained (i.e. $z_a = 1$, imposing the delay propagation), $u_a$ is greater or equal to the delay at the last node of path $a$ (equal thanks to the objective function). The same constraint is redundant if $z_a = 0$, so that $u_a$ assumes the value of zero, thanks to constraint (6) and the objective function. Constraint (5) says that the delay at a node is always greater or equal to zero.

In [14], Schoebel presents the following linear mixed integer problem for the DMP:

\[
\text{(MILP)} \quad \min \sum_{a \in A} p_a [(1 - z_a)T + u_a]
\]
s.t.:

\[ d_{arr,i}^k = D \]  \hspace{1cm} \text{(8)}

\[ d_{arr,i}^k \leq d_{dep,i}^k + slack_i^k \quad \forall i \in V, k \in S_i \]  \hspace{1cm} \text{(9)}

\[ d_{dep,i}^k \leq d_{arr,i}^k + slack_i^{kl} \quad \forall i \in V, l \in S_i : (k, l) \in R_i \]  \hspace{1cm} \text{(10)}

\[ M_{ij}^k(z_a - 1) \leq d_{dep,j}^k + slack_{ij}^k - d_{arr,i}^k \]  \hspace{1cm} \forall a \in A, (i, j, k) \in C_a \]  \hspace{1cm} \text{(11)}

\[ u_a \geq d_{arr,i_a}^k - T(1 - z_a) \quad \forall a \in A \]  \hspace{1cm} \text{(12)}

\[ d_{arr,i}^k \geq 0 \quad \forall i \in V, k \in S_i \]  \hspace{1cm} \text{(13)}

\[ d_{dep,i}^k \leq T \quad \forall i \in V, k \in S_i \]  \hspace{1cm} \text{(14)}

\[ u_a \geq 0 \quad \forall a \in A \]  \hspace{1cm} \text{(15)}

\[ z_a \in \{0, 1\} \quad \forall a \in A \]  \hspace{1cm} \text{(16)}

where \( M_{ij}^k = slack_{ij}^k + T \).

\( d_{dep,i}^k \) is the departure delay of vehicle \( i \) from station \( k \), and \( d_{arr,i}^k \) is the arrival delay of vehicle \( i \) to station \( k \). \( i_a \) is the last vehicle on path \( a \), and \( k_a \) is the last station on path \( a \). Furthermore, the slack times are \( slack_i^k \) for the stop time of vehicle \( i \) at station \( k \), \( slack_i^{kl} \) for the trip of vehicle \( i \) from station \( k \) to the next one \( l \), and \( slack_{ij}^k \) for the change from vehicle \( i \) to vehicle \( j \) at station \( k \).

We show that models (MILP) and (MILP1) are equivalent, i.e. they are both valid formulations of the DMP and their optimal values are equal. Indeed, let \( FS \) be the set of all feasible solutions of (MILP), and \( FS1 \) be the set of all feasible solutions of (MILP1). Then \( FS1 \) is the projection of \( FS \) on the subspace
of variables $z_a, u_a, d_{arr,i}^k : i \in V, k \in R_i, a \in A$.

**Proposition 1** Model (MILP1) is equivalent to model (MILP).

**Proof 1**

As shown in [15], constraints (14) are not necessary. Indeed, if we suppress them, $d_{dep,i}^k$ is no more bounded. But the $d_{dep,i}^k$ are used to determine the values of the $d_{arr,i}^k$ variables, and thus the values of the $d_{arr,i_a}^k$ variables. As those last variables occur in the objective function we want to minimize, we ever choose the $d_{dep,i}^k$ values as small as possible. Since $d_{arr,i}^1 = D < T$, the constraints (9), (10) and (11) from (MILP) and the above argumentation show that we always choose the $d_{dep,i}^k$ values such that $d_{dep,i}^k < T$.

We show how we can suppress the variables $d_{dep,i}^k$ by projecting the (MILP) feasible domain on the subspace of the $z_a, u_a, d_{arr,i}^k : a \in A, i \in V, k \in S_i$ variables. By constraints (9) and (10) of (MILP), we know that:

$$d_{arr,i}^k - slack_i^k \leq d_{dep,i}^k \leq d_{arr,i}^k + slack_i^{kl}$$

$$\forall i \in V, k, l \in S_i : (k, l) \in R_i.$$  

Thus we also have:

$$d_{arr,i}^k - slack_i^k \leq d_{arr,i}^l + slack_i^{kl}$$

$$\forall i \in V, k, l \in S_i : (k, l) \in R_i.$$
In the same way, by constraints (10) and (11), we have:

\[ d_{k, arr,j}^{i} - slack_{j,i}^{k} + M_{ji}^{k}(z_{a} - 1) \leq d_{arr,i}^{k} + slack_{kl}^{i} \]

\[ \forall a \in A, (j, i, k) \in C_{a}, l \in S_{i} : (k, l) \in R_{i}. \]

By applying the Fourier-Motzkin principle (see e.g. [13]), we obtain the following model:

\[(\text{MILP'}) \quad \min \sum_{a \in A} p_{a}[(1 - z_{a})T + u_{a}] \]

s.t.:

\[ d_{arr,1}^{1} = D \quad (17) \]
\[ d_{arr,i}^{k} - slack_{i}^{k} \leq d_{arr,i}^{k} + slack_{kl}^{i} \]
\[ \forall i \in V, k, l \in S_{i} : (k, l) \in R_{i} \quad (18) \]
\[ d_{arr,j}^{k} - slack_{j,i}^{k} + M_{j,i}^{k}(z_{a} - 1) \leq d_{arr,i}^{k} + slack_{kl}^{i} \]
\[ \forall a \in A, (j, i, k) \in C_{a}, l \in S_{i} : (k, l) \in R_{i} \quad (19) \]
\[ u_{a} \geq d_{arr,i}^{k} - T(1 - z_{a}) \quad \forall a \in A \quad (20) \]
\[ d_{arr,i}^{k} \geq 0 \quad \forall i \in V, k \in S_{i} \quad (21) \]
\[ u_{a} \geq 0 \quad \forall a \in A \quad (22) \]
\[ z_{a} \in \{0, 1\} \quad \forall a \in A \quad (23) \]

As (MILP') is the projection of (MILP), their optimal values are identical.

We can easily see that each variable \( d_{arr,i}^{k} \) corresponds to a variable \( d_{n} : n \in N \) (see above). Moreover,
let the \( slack_{ij} : (i, j) \in E \) be equal to \( slack_i^k + slack_i^{kl} \) for the \((i, j)\) in \( arcDA \), and to \( slack_{ij}^k + slack_{ij}^{kl} \) for the \((i, j)\) in \( arcC \). If we change those notations in the \((\text{MILP}')\), we obtain model \((\text{MILP1})\).

As \((\text{MILP1})\) does not consider departure delay variables, we also have to show that there exists an optimal solution of \((\text{MILP1})\) in which the vehicles never leave a station before the scheduled departure time, which was ensured in \((\text{MILP})\) by constraints \((13)\). By contradiction, we suppose that, in an optimal solution, a vehicle has to leave a station before the scheduled depart time. Even if the slack times are zero, this means that the vehicle arrives at the next station before the scheduled arrival time, and that we will have a variable \( d_i < 0 \). But, this is in contradiction with constraint \((5)\) from \((\text{MILP1})\).

\[ \square \]

### 3 Reducing the number of variables

Here we propose another model which involves variables \( z_a \) and \( u_a \) only. Consider the following model:

\[
\text{(MILP2) } \min \sum_{a \in A} p_a [(1 - z_a)T + u_a]
\]

s.t.:

\[
Tz_a' - u_a' + \sum_{(i,j) \in arcC_a} M_{ij}z_{a''(i,j)} \\
\leq T - D + \sum_{(i,j) \in E_a} slack_{ij}^k + \sum_{(i,j) \in arcC_a} M_{ij},
\]

\( \forall a, a', a''(i, j) \in A \) so that the first node on path

\[ a \text{ is } 1, v_{a'} = v_a \text{ and } (i, j) \in arcC_a \cap arcC_a'' \quad (24) \]

\[ u_a \geq 0 \quad \forall a \in A \quad (25) \]

\[ z_a \in \{0, 1\} \quad \forall a \in A, \quad (26) \]
where $E_a$ is the set of arcs on path $a$.

We show that models (MILP1) and (MILP2) are equivalent. Indeed, we suppress the arrival delay variables from (MILP1) by projecting the polyhedron associated to the LP-relaxation of (MILP1) on the space of the other variables $z_a, u_a : a \in A$ of the problem. We then obtain (MILP2).

**Proposition 2** Let $FS_1$ be the (MILP1) feasible solutions set, and $FS_2$ the (MILP2) feasible solutions set. Then $FS_2$ is the $FS_1$ projection on the subspace of the $z_a, u_a : a \in A$ variables, that is to say: $FS_2 = \{(z_a, u_a) \text{ s.t. there exists } d_i \text{ with } (z_a, u_a, d_i) \in FS_1\}$.

**Proof 2**

Starting from the (MILP1), we apply Fourier-Motzkin elimination to $d_i : i \in N$ variables. In other words we project the (MILP1) feasible domain on the subspace of the $z_a, u_a : a \in A$ variables.

For each arc $(i, j)$ from graph $G = (N, E)$, we define:

$$l_{ij} = \begin{cases} 
\min_{a \in A : (i, j) \in arcC_a} \left( slack_{ij} + M_{ij}(1 - z_a) \right), \\
\text{if } (i, j) \in arcC \\
slack_{ij}, \quad \text{if } (i, j) \in arcDA
\end{cases}$$

These weights on the arcs allow us to evaluate the delay at each node. Indeed, if $(i, j) \in arcDA$, we have, thanks to constraint (2):

$$d_j \geq d_i - slack_{ij}.$$ 

In the same way, if $(i, j) \in arcC$, constraint (3) says:

$$d_j \geq d_i - slack_{ij} - M_{ij}(1 - z_a) \quad \forall a \in A : (i, j) \in arcC_a,$$
Thus we have:

\[ d_j \geq d_i - \min_{a \in A; (i,j) \in \text{arc}_a} (\text{slack}_{ij} - M_{ij}(1 - z_a)). \]

Consequently, for each arc \((i, j)\) of the graph, we have:

\[ d_j \geq d_i - l_{ij}. \]

Let us now concentrate on a path \(a\) whose first node is 1 (i.e. the node where the initial delay occurs), and suppose its nodes are called 1, 2, ..., \(v_a\). We know, from (MILP1), that \(d_1 = D\) and \(d_2 \geq d_1 - l_{12}\). If we put together these two inequalities, we obtain:

\[ d_2 \geq D - l_{12}. \]

As we also know that \(d_3 \geq d_2 - l_{23}\), we have:

\[ d_3 \geq D - l_{12} - l_{23}. \]

If we continue like this up to \(d_{v_a}\), and if we call \(E_a\) the set of arcs on path \(a\), we obtain:

\[ d_{v_a} \geq D - \sum_{(i,j) \in E_a} l_{ij}. \]

It is clear that these arguments are valid for all paths \(a\) whose first node is 1. In addition, it is not necessary to consider other paths, as all the paths are included in \(A\) and the delay at the end of those other paths \(a'\) is either zero or the same delay as the one at the end of a path whose first node is 1. Indeed, suppose we have a path \(a'\) such that \(v_{a'} = v_a\) and whose first node is not 1. Two situations can happen:
• if there is no intersection between this path $a'$ and paths whose first node is 1, then the delay at each node of path $a'$ is zero;

• if the path $a'$ has one (or more) intersection with at least one path whose first node is 1, then:
  
  – the $a'$ nodes which are before the first intersection do not have any delay;
  
  – the delay at the other $a'$ nodes are the same as those on paths whose first node is 1.

Thus the following inequality:

$$d_{v_{a}} \geq D - \sum_{(i,j) \in E_{a}} l_{ij} \forall a \in A \text{ whose first node is 1} \quad (27)$$

can replace constraints (2) and (3).

Furthermore, if we put together inequality (27) and constraint (4), we have:

$$D - \sum_{(i,j) \in E_{a}} l_{ij} \leq d_{v_{a}} \leq T(1 - z_{a'}) + u_{a'}$$

for each $a, a' \in A$ so that a first node is 1 and $v_{a'} = v_{a}$.

By applying Fourier-Motzkin elimination, the following inequality replaces constraints (27) and (4):

$$D - \sum_{(i,j) \in E_{a}} l_{ij} \leq T(1 - z_{a'}) + u_{a'}$$

$\forall a, a' \in A$ so that a first node is 1 and $v_{a'} = v_{a}$. \quad (28)

Summarizing, we obtain the following model:
\[ \min \sum_{a \in A} w_a [(1 - z_a)T + u_a] \]

s.t. :

\[ D - \sum_{(i,j) \in E_a} l_{ij} \leq T(1 - z_a') + u_a' \]

\( \forall a, a' \in A \) so that a first node is 1 and \( u_a' = v_a \).

(29) \[ u_a \geq 0 \quad \forall a \in A \] (30)

\[ z_a \in \{0, 1\} \quad \forall a \in A. \] (31)

Let us now replace the terms \( l_{ij} \) using their definitions. First, note that:

\[
\min_{a \in A; (i,j) \in \text{arc}_a} (\text{slack}_{ij} + M_{ij}(1 - z_a))
\]

\[ = \text{slack}_{ij} + M_{ij} - \max_{a \in A; (i,j) \in \text{arc}_a} M_{ij} z_a, \]

and let us define \( \tilde{z}_{i,j} = \max_{a \in A; (i,j) \in \text{arc}_a} z_a \). Then, for each arc \( (i,j) \in \text{arc}_C \), we have:

\[ l_{ij} = \text{slack}_{ij} + M_{ij} - M_{ij} \tilde{z}_{i,j}. \]

Thus we can write the model above as:

\[ \min \sum_{a \in A} w_a [(1 - z_a)T + u_a] \]
\[ T z_{a'} - u_{a'} + \sum_{(i,j) \in \text{arc}_a} M_{ij} \tilde{z}_{ij} \leq T - D + \sum_{(i,j) \in E_a} \text{slack}_{ij} + \sum_{(i,j) \in \text{arc}_a} M_{ij} \]

\( \forall a, a' \in A \) so that a first node is 1 and \( v_{a'} = v_a \). \hspace{1cm} (32)

\( u_a \geq 0 \ \forall a \in A \) \hspace{1cm} (33)

\( z_a \in \{0, 1\} \ \forall a \in A. \) \hspace{1cm} (34)

Unfortunately, because of the terms \( \tilde{z}_{i,j} \), this model is not linear. However, as \( \tilde{z}_{i,j} \) is defined as a maximum \( (\tilde{z}_{i,j} = \max_{a'' \in A; (i,j) \in \text{arc}_{a''}} z_{a''}) \), the inequality still holds if we replace, in each term of the sum, \( \tilde{z}_{i,j} \) by any \( z_{a''} : (i, j) \in \text{arc}_{a''} \). As those last \( a'' \) depend on the term \( (i, j) \) in the sum, we denote them \( a''(i, j) \).

Consequently we can transform the model above to obtain (MILP2).\( \square \)

**Corollary 1** The (MILP1) and (MILP2) optimal values are identical.

It is interesting to note that model (MILP2) has the following interpretation. Focussing on path \( a' \) and on variable \( u_{a'} \), constraint (24) can be written as:
\[ u_{a'} \geq T(z_{a'} - 1) + D - \sum_{(i,j) \in E_a} slack_{ij} \]
\[ + \sum_{(i,j) \in \text{arc}C_a} M_{ij}(\tilde{z}_{i,j} - 1) \]
for all paths \( a \) s.t. the first node of \( a \) is 1 and \( v_{a'} = v_a \).

If \( z_{a'} = 0 \), the constraint is redundant, and the contribution of path \( a' \) in the objective function is given by \( T \).

If \( z_{a'} = 1 \), we have:

\[ u_{a'} \geq D - \sum_{(i,j) \in E_a} slack_{ij} + \sum_{(i,j) \in \text{arc}C_a} M_{ij}(\tilde{z}_{i,j} - 1) \]
for all paths \( a \) s.t. the first node of \( a \) is 1 and \( v_{a'} = v_a \).

We consider a “primary network” given by the path \( a \) and we refer the delay on \( a' \) to this primary network. Two situations can happen:

- If all connections on the primary network are maintained, the delay for \( a' \) passengers is the same as that for \( a \) passengers (we recall that, by definition of the Arr-Arr graph, having the same destination node implies arriving at the same station with the same vehicle). Indeed, in this case \( \sum_{(i,j) \in \text{arc}C_a} M_{ij}(\tilde{z}_{i,j} - 1) = 0 \) causing \( u_{a'} \geq D - \sum_{(i,j) \in E_a} slack_{ij} \), which is the delay at path \( a \) destination.

- On the contrary, if at least one connection on the primary network is not maintained, then path \( a \)
does not have any influence on the delay on \( a' \). Hence, the constraint is redundant.

## 4 Application

Model (MILP1) has been implemented using AMPL (see [6]) and solved using Cplex 8.1 (see [11]), as well as model (MILP).

As model (MILP2) contains an exponential number of constraints, a direct implementation is not appropriate and a constraint generation approach is proposed. We start solving the (MILP2) using only constraints (25) and (26). Generally, the solution obtained will not be feasible, as some constraints (24) might be violated. We then select a subset of the violated constraints and we add them to the model. The constraints adding procedure is repeated, until a solution is found which does not violate any constraints: this solution is also optimal for (MILP2).

For the sake of the constraint generation procedure, it is sufficient to generate inequalities (32) instead of inequalities (24). From (32), given the optimal solution \( z^*, u^* \) of the current model, a violated inequality exists if

\[
D - \sum_{(i,j) \notin \text{arcC}_a} \text{slack}_{ij} - \sum_{(i,j) \in \text{arcC}_a} [\text{slack}_{ij} + M_{ij}(1 - z_{i,j}^*)] > T(1 - z_{a'}^*) + u_{a'}^*
\]

for some \( a, a' \in A \) so that \( a \) first node is 1 and \( v_{a'} = v_a \). By definition of \( l_{ij} \), this corresponds to
\[ D = \sum_{(i,j) \in E_a} l_{ij} > T(1 - z^*_a) + u^*_a \] 

The term \( d_{v_a} = D - \sum_{(i,j) \in E_a} l_{ij} \) represents the delay at the last node of path \( a \), for the given \( z^* \) and can be evaluated using a breath-first visit of the Arr-Arr graph (delay propagation on acyclic network).

The separation procedure computes the node delays and then checks for a path satisfying condition (35), as from the following procedure.

1. For each \((i,j) \in arcC\), determine the variable \( z^*_{i,j} \).
2. For each node of the Arr-Arr graph \( v \in V \), compute the delay \( d^*_v \) according to the values \( u^* \) and \( \bar{z}^* \) (breath-first visit of the Arr-Arr graph).
3. For each path \( a \in A \) whose first node is 1
4. For each path \( a' \) whose last node is \( v_{a'} = v_a \)
5. If \( d^*_{v_a} > T(1 - z^*_a) + u^*_a \), then inequality (32) is violated for \( a, a' \).
6. Go to the next path \( a' \).
7. Go to the next path \( a \).

Step 1 runs in \( O(|arcC||A|) \) time: for each connection, all the paths involving the connection itself are considered. Step 2 consists of a breath first visit of the Arr-Arr graph, whose complexity is \( O(|E|) \). Steps from 3 to 7 run in \( O(|A|^2) \). In terms of the DMP input, i.e. the vehicle set \( V \), the station set \( S \) and the passenger path set \( A \), we can state \( |arcC| = O(|V|^2|S|) \) and \( |E| = O(|S|^2 + |V|^2|S|) \). It follows that the overall complexity of the separation procedure is \( O(|V|^2|S||A| + |S|^2 + |A|^2) \) and thus polynomial.
Table 1: Instance summary.

| $N$ | $|arcdA|$ | $|arcC|$ | $|A|$ | Num. |
|-----|--------|--------|-------|------|
| 16 - 20 | 11 - 16 | 3 - 11 | 62 - 132 | 28 |
| 25 - 40 | 20 - 33 | 3 - 14 | 99 - 549 | 88 |
| 59 - 101 | 55 - 93 | 10 - 22 | 1821 - 6366 | 12 |

Table 2: Computational results.

<table>
<thead>
<tr>
<th>1. Inst.</th>
<th>2. $D$</th>
<th>3. MILP1</th>
<th>4. MILP2</th>
<th>5. $%$ sep.</th>
<th>6. MILP1</th>
<th>7. MILP2</th>
<th>8. $%$ sep.</th>
<th>9. $%$ z=1</th>
<th>10. p(z=0)</th>
<th>11. $%$ z=0</th>
<th>12. gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-20</td>
<td>12</td>
<td>0.00</td>
<td>0.04</td>
<td>9.1%</td>
<td>0.01</td>
<td>0.01</td>
<td>8.0%</td>
<td>98.33%</td>
<td>16.67%</td>
<td>28.33%</td>
<td>11.67%</td>
</tr>
<tr>
<td>16-21</td>
<td>20</td>
<td>0.00</td>
<td>0.05</td>
<td>12.3%</td>
<td>0.04</td>
<td>0.05</td>
<td>9.0%</td>
<td>96.17%</td>
<td>33.33%</td>
<td>33.33%</td>
<td>11.67%</td>
</tr>
<tr>
<td>16-22</td>
<td>25</td>
<td>0.00</td>
<td>0.05</td>
<td>13.2%</td>
<td>0.03</td>
<td>0.05</td>
<td>9.0%</td>
<td>96.17%</td>
<td>33.33%</td>
<td>33.33%</td>
<td>11.67%</td>
</tr>
<tr>
<td>25-40</td>
<td>12</td>
<td>0.00</td>
<td>0.05</td>
<td>13.6%</td>
<td>0.03</td>
<td>0.05</td>
<td>9.0%</td>
<td>96.17%</td>
<td>33.33%</td>
<td>33.33%</td>
<td>11.67%</td>
</tr>
<tr>
<td>25-41</td>
<td>20</td>
<td>0.00</td>
<td>0.06</td>
<td>14.6%</td>
<td>0.02</td>
<td>0.06</td>
<td>9.0%</td>
<td>96.17%</td>
<td>33.33%</td>
<td>33.33%</td>
<td>11.67%</td>
</tr>
<tr>
<td>25-42</td>
<td>25</td>
<td>0.00</td>
<td>0.03</td>
<td>1.4%</td>
<td>0.05</td>
<td>0.04</td>
<td>1.0%</td>
<td>98.52%</td>
<td>4.76%</td>
<td>31.00%</td>
<td>11.67%</td>
</tr>
<tr>
<td>25-43</td>
<td>28</td>
<td>0.00</td>
<td>0.03</td>
<td>1.4%</td>
<td>0.05</td>
<td>0.04</td>
<td>1.0%</td>
<td>98.52%</td>
<td>4.76%</td>
<td>31.00%</td>
<td>11.67%</td>
</tr>
<tr>
<td>55-101</td>
<td>12</td>
<td>0.15</td>
<td>0.14</td>
<td>76.0%</td>
<td>0.45</td>
<td>0.46</td>
<td>75.5%</td>
<td>100.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>55-102</td>
<td>20</td>
<td>0.15</td>
<td>0.14</td>
<td>76.0%</td>
<td>0.44</td>
<td>0.46</td>
<td>75.5%</td>
<td>100.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>55-103</td>
<td>25</td>
<td>0.15</td>
<td>0.14</td>
<td>76.0%</td>
<td>0.44</td>
<td>0.46</td>
<td>75.5%</td>
<td>100.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>55-104</td>
<td>28</td>
<td>0.15</td>
<td>0.14</td>
<td>76.0%</td>
<td>0.44</td>
<td>0.46</td>
<td>75.5%</td>
<td>100.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

At each iteration of the constraint adding procedure, all the violated inequalities generated by step 5 are added (computational results put in evidence that this is better than adding a subset of violated constraints).

Note that the constraint generation approach can be applied to both (MILP2) and its linear relaxation. In the former case, we obtain the optimal solution to the DMP, in the latter case the procedure should be integrated in a Branch and Bound framework.

In order to show the applicability of the proposed models to real contexts, the AMPL implementation of (MILP1) and the constraint generation approach for (MILP2) have been tested on the instances described in Table 1. Those instances were derived as subnetworks of the intermodal public transportation network around Brussels in Belgium. In particular, we consider a subset of the medium frequency lines connecting Brussels with some of the most important towns in the “Brabant wallon” region. The average frequency is one vehicle each 30 minutes. Both the connections of these lines with each others and with the intermodal public transportation network of Brussels (including metro, tram and bus lines) are taken into account, as they emerge by the timetable and have been confirmed by line operators.
Three classes of instances have been identified, depending on the size of related Arr-Arr graph. For each class, the minimum and the maximum cardinality of the set describing the DMP are given, together with the number of instances, in the last column. The slack times have been randomly generated as integer numbers between zero and three minutes. Initial delays $D$ of 12, 20, 25 and 28 minutes have been considered for each instance and a time horizon $T$ of 30 minutes. The number of passengers on each path has been obtained as follows: first we have randomly generated the origin/destination matrix; after that, we have used a logit function (see e.g. [5]) to distribute any entry of the matrix among the paths having the same origin and the same destination, this function taking into account the number of connection arcs and the total length of each path.

The results obtained on a 3 GHz Pentium IV processor are summarized in Table 2. The first two columns give the instance size and the initial delay. The third and the fourth columns give the time spent to solve the linear relaxation of (MILP1) and (MILP2). Column 5 shows the percentage of time spent by the separation procedure. Columns 6, 7, 8 give the same information with respect to the original (MILP1) and (MILP2), including the integrality constraints. Columns from 9 to 11 give some statistics on the solution values: column 9 gives the percentage of maintained connections, column 10 refers to the percentage of instances containing suppressed connections and column 11 reports the percentage of suppressed connection within those instances. The last column show the integrality gap.

Both Cplex directly applied to (MILP1) and the constraint generation procedure applied to (MILP2) are able to solve the instances in a very reasonable amount of time: less than one second, both for the linear relaxation and the integer program. Indeed, Cplex opens a very small number of branch and cut nodes when applied to (MILP1) and often closes the gap at the root node, using the built-in cuts. Also, six constraint generation iterations are sufficient in average to solve the linear relaxation of (MILP2), and four to solve the related integer program (we recall that in this case the Cplex branch and cut procedure
Figure 3: Performance evaluation: linear relaxation.

Figure 4: Performance evaluation: integer program.
is applied at each iteration).

A simple comparison between the performance of the two proposed models and related methods is highlighted in Figures 3 and 4: they show, for different instance classes, the frequency one method has been able to find the optimal solution faster (or within the same computational time) than the others. We can observe that MILP1 performs better for the small instances we considered, while the constraint generation approach tends to be better with instances of increasing size, despite of the fact that the the majority of the time is spent by the separation procedure. Actually, just a few iterations provide the optimal (relaxed or integer) solution: a very small subset of constraints (24) is added and considerably smaller problems have to be solved.

Concerning the comparison with model (MILP), its application has been limited to preliminary computational experiments, including instances of smaller size. Actually (MILP), (MILP1) and (MILP2) formulations provide the same LP-relaxation optimal value and the only difference is in term of computational times. This difference cannot be appreciated for the considered instances, due to the very small processing times. Indeed, the formulation of (MILP) is almost the same as the one for (MILP1), except that it has a larger number of variables (corresponding to the departure delay variables) and some additional constraints (corresponding to the redundant constraint we have suppressed). As a consequence, we do not expect (MILP) to perform better than (MILP1).

Concerning the value of the optimal solutions, we observe that, in most cases, all the connections are maintained. Moreover, in the instances where some connections are suppressed, it only concerns a small number of paths (from 10 to 30%, depending on the amount of the initial delay).
5 Conclusions

This paper has presented two new MILP formulations for the Delay Management Problem. The first formulation (MILP1) is based on a new graph interpretation of the DMP, which allows us to reduce the number of variables and constraints with respect to equivalent models presented by literature. The second equivalent model (MILP2) further reduces the number of variables, by projecting out all the variables explicitly related to node delays. The cost to pay is the increasing number of constraints, which are exponentially many. The computational results presented in this paper show that the trade off between the number of variables and the number of constraints tends to privilege (MILP2) and the related constraint generation procedure when facing instances of greater size, whose magnitude is similar to real public transportation networks. This suggests to further improve the solution approach based on (MILP2), in particular by enhancing the separation procedure (which takes most of the computational time and is presently based on path enumeration) and by integrating it in a Branch and Cut context.

Also note that, at the moment, models (MILP1) and (MILP2) assume that the frequency of any line in the public transport network is the same, so that a single parameter \( T \) is considered to represent the period of those lines. Real networks usually present different periods, for different lines: for example, if we consider the intermodal public transportation network in Brussels, the period for metro lines during off-peak hours is 10 minutes, while the period of trams or busses is 20 minutes on average. Assuming a single parameter \( T \) could be misleading: future work will be to refine the initial assumption and to differentiate the parameter \( T \) per path.

From the public transportation network point of view, the results show that there is room for improving the connection protocols currently adopted by the carriers. In fact, the decisions about the suppression of a connection is mostly based on a threshold delay of the incoming vehicles. Results show that, at least from the passengers point of view, this is not always the best decision: most of connection should
be maintained even under large vehicle delays, as this leads to smallest total passenger delays, thus considerably increasing the attractiveness of public transportation networks.

Acknowledgements. The authors wish to thank one anonymous referee for her/his helpful comments.

References


