Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 10 (2015) 483 – 492

Transportation Research Procedia

www.elsevier.com/locate/procedia

18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands

Delay management in public transportation: service regularity issues and crew re-scheduling

S. Carosi [a], S. Gualandi [b], F. Malucelli [c], E. Tresoldi [c,*]

[a] *M.A.I.O.R., Via San Donato 512, 55100 Lucca, Italy*
[b] *AntOptima, Via Aprica 26, 6900 Lugano, Switzerland*
[c] *Politecnico di Milano - Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza L. da Vinci 32, 20133 Milano, Italy*

Abstract

In this paper, we propose a decision support tool to assist a local public transportation company in tackling service delays and small disruptions. We discuss different ways to assess and improve the regularity of the service, and we propose a simulation based optimization system that can be effectively used in a real-time environment taking into account both vehicle and driver shifts. In particular, we describe a tabu-search procedure for the online vehicle scheduling optimizing the regularity of the service and a column generation approach for the consequential crew re-scheduling minimizing the driver extra-time. As a case study, we analyze the management of urban surface lines of Azienda Trasporti Milanese (ATM) of Milan. In the last part of the paper we report a detailed analysis of the experimental phase showing the effectiveness of the proposed approach.

Peer-review under responsibility of Delft University of Technology

Keywords: Delay Management, Transportation, Re-optimization, Real-time Optimization, Real-word scenario

1. Introduction

The quality of a local public transport depends on the perceived efficiency and reliability. However, events or disturbances innate in the system, especially in an urban setting, may generate disruptions that negatively influence this perception. Additionally, disruptions usually increase the operating cost, for instance, involving extra allowances for bus drivers, or penalties to be paid to the municipality that commended the service. In this paper we focus on day-to-day situations where the regularity of the service is compromised by delays and small disruptions. Currently, the daily operations of transit companies are monitored "manually" by an operation central office taking advantage of Automated Vehicle Monitoring (AVM) systems and mobile telecommunication devices. Each operator controls the operations of one or more lines, detecting delays that may generate disruptions or collecting information about

* Corresponding author.
E-mail address: emanuele.tresoldi@polimi.it

2352-1465
Peer-review under responsibility of Delft University of Technology
doi:10.1016/j.trpro.2015.09.002

problems on the line such as vehicle breakdowns or accidents. In the presence of a disruption, the operator assists remotely the drivers deciding the actions to be taken in the line. In addition, the operator must inform passengers both onboard and waiting at the stops, of the new solution adopted to tackle the disruption. The basic actions that an operator in the central office can enforce are, for instance, to perform vehicle detours, to delay a vehicle schedule, or to cancel one or more trips. The operator may also decide to use spare resources (drivers or vehicles). However, this is not an option for the ordinary disruption management since, usually, these resources are extremely scarce and are utilized in exceptional cases only.

Methods to cope with disruptions have been introduced initially in the airline industry (see for example Clausen et al. (2010)) and then extended to railways Jespersen-Groth et al. (2009), Cacchiani et al. (2014). The complexity of the constraints emerging in railways, mainly due to the shared infrastructure, often suggested hybrid approaches conjugating optimization techniques with simulation Berger et al. (2011). Only recently this problem has been tackled in local public transport. In this context, peculiar features motivate an ad hoc study of disruption management methods that can take advantage of transit additional flexibility such as trip cancellations, detour or limitations that add significant degrees of freedom and open for different types of approaches. For example Li et al. (2009) cope with the vehicle re-scheduling problem in the presence of a disruption due to the breakdown of a single vehicle and other vehicles must deviate in order to take care of passengers. This kind of problem appears suited for extra urban transportation where the frequency is low, rather then for the urban case. Bartholdi-III and Eisenstein (2012) tackle the problem of equally spacing buses in a urban environment. This paper however does not consider the issues deriving from the modified driver scheduling.

In this paper we present a new simulation based optimization algorithm that can assist the operators to face different types of small delays and disturbances with the ultimate objective of increasing the quality of service, or at least to limit the perception of inconvenience on passengers, using the available resources. After a short analysis of different methods for the evaluation of the regularity of the service (Section 2), we introduce the simulation procedure and we define the *delay managment algorithm* (Section 3). We describe the tabu-search algorithm employed for the real-time re-scheduling of the vehicles (Section 3.1) and the column generation method tackling the crew scheduling re-optimization (Section 3.2). Finally, we report the results obtained by our procedure in real-world scenarios arising in the urban management of surface lines of Azienda Trasporti Milanese (ATM) of Milan (Section 4) and we draw some conclusions depicting future developments (Section 5).

2. Evaluating the regularity of the service

The main challenge in managing disturbances in the regularity of the service is to be able to distinguish events that may have a negative impact on the service from those whose effect will recover in a short time without interventions.

In order to develop an algorithm that proposes a course-of-actions to mitigate the effect of disturbances, we need a precise and formal way to evaluate the regularity of the service along one line. The public transportation lines that we are taking into account provide a *frequency based service*: at each stop a frequency of vehicle passages is specified, instead of the timetable (e.g., a vehicle every 7 minutes). In this case, even if generalized delays are present, the service is not perceived as disrupted by passengers, provided that frequencies are regular and aligned with the planned ones. Note that, in this type of service, even though users perceive the service delivered according to frequencies, resources, namely vehicles and drivers, continue to be managed on a timetable base, hence delays or other events which are not perceived as disruptions by users, may generate disruptions when the resource management must be inevitably adjusted. These disruptions are called endogenous.

There are many ways to estimate the regularity of the service. The most regular service is obviously that reproducing exactly the planned timetable. Thus the regularity measure should consider the adherence of the provided service with the planned one. In the case of timetabled service the measure will consider the planned timetable, while in the frequency based service this measure can be relaxed and only the headways will be accounted for. In the literature many proposals are present (see for example Barabino et al. (2013) for a brief survey). It can be observed that, most of the times, these measures are suitable when computed off line, when all data collected during the day are available. This is particularly true when the average, the standard deviation or other cumulative indicators of the observed value are used. However, this is not always possible when the measure has to be computed in real time to support decisions about disruption detection and possible recovery actions. In the following we will propose some functions to be used to measure the regularity in real time.

For the sake of simplicity let us consider a circular line with n total stops. Our definition can be extended to more involved cases. Let n_p be the number of planned stops within a whole day of service, that is the number of stops of the line in both directions multiplied by the number of trips.

2.1. Headway based index of regularity: a simple 0-1 proposal

Let n_b be the number of times a vehicle arrives later than a given threshold t with respect of the planned headway, these are called the *bad passes*. The index of regularity of the line is computed as:

$$\left(\frac{n_p - n_b}{n_p} \times 100\right) \tag{1}$$

and expressed as percentage of bad passes over the total number of planned passes n_p. This index of regularity based on a simple 0-1 penalty function is currently used by ATM for internal purposes.

Notice that this type of index does not account for early passes. This feature is intentional, since, considering the actual headway, a vehicle anticipating its pass at a stop, while the other vehicles are maintaining their schedule, will result as an early pass with respect to the previous vehicle, but the following vehicle will result as having a bad pass considering the actual headway. Thus penalizing early passes as well as late ones (of the same amount), would penalize twice the same event. However, this type of index of regularity penalizes in the same way an almost good pass and a very bad one and may generate some pathological behaviors, as illustrated by the following example.

Example 1. *Consider the situation in Figure 1(a) where we have all vehicles grouped in a pack: the first (blue) bus is late, while the following four (red) buses are ahead of schedule. With the current index, only the blue bus originates a bad pass, and the regularity of service will be around 80%, which, in general, is sufficiently good. Nevertheless, the situation depicted in Figure 1(a) is clearly unacceptable. In addition, if the operators try to improve the regularity by detouring a (red) vehicle of the pack, as shown in Figure 1(b), the average waiting time of users decreases, but the index of regularity worsens since the detoured vehicle would generate new bad passes, which is clearly a paradox. Hence, the use of such an index in an automated system that seeks to improve the index of regularity, in the presence of disturbances, would make the service converge towards situations as that depicted in Figure 1(a), and from where it is impossible to escape by applying recovery actions to one vehicle at a time.*

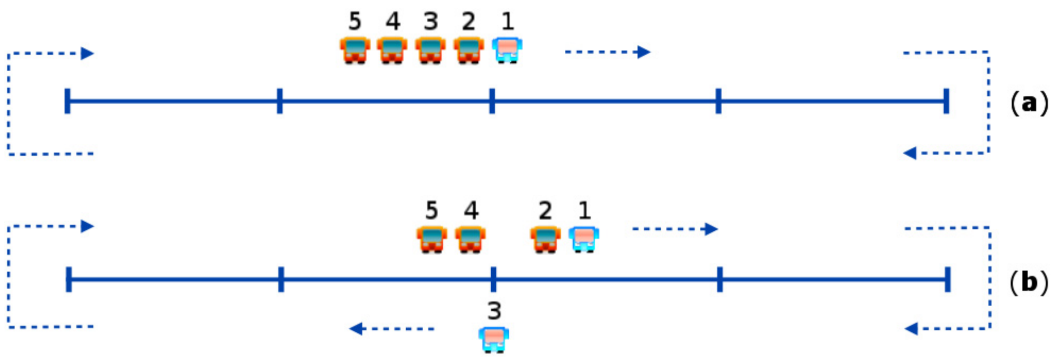

Fig. 1. (a) The first bus is late; the other four buses are travelling ahead of schedule. (b) The current index of regularity will decrease if the 3-rd bus were to turn at mid-line.

2.2. Piecewise linear function for evaluating the index of regularity

In order to overcome the drawbacks emerging from the previously described method for evaluating the regularity, we can consider a function $f(x)$ of the gap x between the planned headway and the observed one. Denoting by h_p the planned headway and by h_o the observed one, the gap can be computed as $x = h_o - h_p$. A negative value of x means an early pass, while a positive value means a delay. The function used to define the index of regularity is:

$$f(x) = \begin{cases} -\alpha x & \text{if } x < -\theta_1 \\ 0 & \text{if } -\theta_1 \leq x < \theta_2 \\ \beta x & \text{if } \theta_2 \leq x < \theta_3 \\ \gamma x + \delta & \text{if } \theta_3 \leq x \end{cases} \tag{2}$$

where $\theta_1, \theta_2, \theta_3 (> \theta_2)$ and $\alpha, \beta, \gamma, \delta$ are suitable parameters. If we want to ignore the contribution of earliness it suffices to set $\alpha = 0$. Note that this index includes also the simple 0-1 index if we set $\alpha = \beta = \gamma = 0$ and $\delta = 1$. The contribution of the function $f(x)$ due to values $x \geq \theta_3$ intends to penalize large gaps more than the equivalent sum of small gaps.

We studied other service regularity measures based on quadratic penalty functions or that, in addition to the headway gap, account for the adherence with respect to the driver scheduling. In our preliminary experiments, we decided to focus on the piecewise penalty function described above.

3. Simulation-based evaluation

Currently the officers take their decisions about recovering the regular service relying upon their expertise. The actions that they can enforce to drivers are basically unplanned detours and changing the duration of breaks at terminals. In certain locations along the line, it is also possible to hold the vehicle, if it is off duty. The effects of these actions on the service regularity are evaluated only according to their experience and intuition, and it is almost impossible to assess the impact of alternatives on the service regularity and on the driver scheduling. However, with the support of an automated system the *estimate* of the effects of every action is possible by means of a stochastic simulation that looks forward into the future, as in Hickman (2001). The simulator considers the empirical distributions of traveling times obtained with an analysis of the historical data Gualandi et al. (2014). Note that when frequency are coarse and the variance of the headway is high, also dwell time play an important role. Having a detailed information about the demand distribution over the time also this aspect can be easily accounted for in the simulation.

3.1. Delay Management Algorithm

We present in this section an algorithm that plans and simulates a course-of-actions to react to delays or minor disruptions. A general overview of the system is shown in Figure 2. The routine is composed of seven steps that are described in the reminder of this section referring, without loss of generality, to as a single line.

Step 1: snapshot of the current status of the line. The current status of the line is obtained from the data generated by the AVM system: considering a starting time t_0, the algorithm collects the AVM data for every active vehicle (last observed position and time) and stores it in the set L.

Step 2: computing index of regularity without actions. A forecast of the evolution of the line is obtained simulating, starting from t_0, the operations of the vehicles for the next n hours. The core of Step 2 is the *AVM simulation routine*, based on a procedure that mimics the AVM data generation from t_0 to $t_0 + n$. The procedure takes as input the current line status L, the vehicle scheduling that is the planned trips and the expected headway of vehicles at the stops, and the crew scheduling from time t_0 to the end of the day. Moreover, it uses a priority queue Q whose elements represent AVM observations (i.e., triples (vehicle, position, time)) and are sorted by increasing time. Q, is initialized by including all elements of L into Q, and an observation for each new vehicle that enters the service in the period $[t_0, t_0 + n]$. Such observations are time-stamped with the expected time when the vehicle should start the service.

In the simulation loop, the first element Ob' is extracted from Q. Starting from Ob' a new fictional AVM observation Ob'' is generated corresponding to the next event that will happen for the vehicle of Ob', that is the arrival of the vehicle at the next stop on its trip or the begining of a new trip. The time-stamp for Ob'' is obtained starting from the

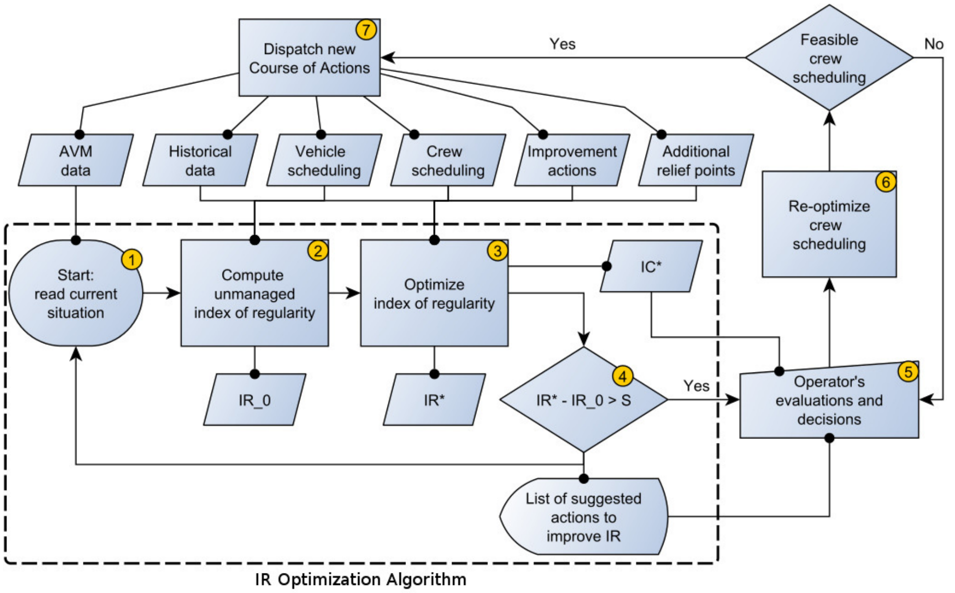

Fig. 2. Delay Management Procedure

Ob' time-stamp and adding the travel/waiting time to reach the stop of Ob''. Such value is predicted considering the historical empirical distributions for the given month, day of the week and time of the day, and the possible influence of the delay on dwell times. Ob'' is added to Q unless its time-stamp exceeds $t_0 + n$. If Ob'' represents the last event for that vehicle in the time period $[t_0, t_0 + n]$ (e.g., vehicle returning to the depot) then Ob'' is discarded. The AVM simulation routine ends when Q is empty. During the simulation the index of regularity is evaluated according to one of the functions described in section 2 to estimate how the regularity of service is expected to evolve. Let us call IR_0 the value of the expected index of regularity in the period $[t_0, t_0 + n]$ when no actions are applied.

Step 3: improving the index of regularity. The following elements are considered: an index of regularity function, called $IR(\cdot)$, the current line status L, the vehicle and crew scheduling data (V, C), a set $\mathcal{A}$ of actions, such as, for instance, detours and shortcuts, with or without the possibility of holding the vehicle, and minimum and maximum holding time at the terminals. The optimization problem solved in this step consists of determining a (limited) subset $\mathcal{B} \subset \mathcal{A}$ of actions, which, if applied, will allow with high probability to improve IR_0 computed in the previous step.

This problem is solved with an algorithm based on tabu-search paradigm (see Glover and Laguna (1997)). In detail: the algorithm starts from the initial status that is the schedule obtained in the Step 2 then for every iteration it moves to the best available status inside its neighborhood (even if it is not improving the current IR); the process stops when the maximum number of iterations is reached. Given a status its neighborhood is composed by all different statuses that can be reached using two different types of moves: *actions taken* and *actions removed*. The former type of move consists in implementing one action among those in A. The latter type of move removes one action that was implemented in a previous iteration bringing the involved trip back to its original structure. In order to let the algorithm escape from local optima we implemented a tabu-search mechanism consisting in a list F of actions that are forbidden and cannot be taken. In particular, in our procedure at every iteration we store in F, for the next f iterations, all the actions associated with the trip modified in the current iteration.

Algorithm 1 reports the pseudo-code of the tabu-search routine. This procedure is executed k times (lines 2 - 16) and in each iteration every action $a \in A \setminus F$ is tested (lines 5 - 10). This is done updating V to V_a in order to take into account all the modifications required by the action a (line 6). Then the simulation algorithm described in Step 2 is executed taking V_a, together with L and C as input and the corresponding index of regularity IR_a is computed along the simulation process (line 7). In every iteration the action a^* leading to the best IR_a is memorized (lines 9 and 10) and is used to update the vehicle scheduling data and the tabu list for the next iteration (lines 11 and 12). The overall

Algorithm 1: IR OPTIMIZATION ALGORITHM

Input: IR_0 the unmanaged index of regularity; A a set of possible actions (e.g., unplanned detours, shortcuts); V vehicles scheduling data (e.g. vehicles timetables, trips, ...); C crews scheduling data (e.g. crews timetable, relief points, ...); L snapshot of the initial situation of the line at time t; F tabu table as list of forbidden actions; k number of simulations

Output: $B \subset A$ a set of possible actions to improve the index of regularity; V^* vehicles scheduling data obtained applying B to V; IR^* index of regularity computed on V^*;

1 $IR^* \leftarrow IR_0, V^* \leftarrow \emptyset, B \leftarrow \emptyset, T \leftarrow \emptyset$;
2 **foreach** $i \in 1, \ldots, k$ **do**
3     $IR_p \leftarrow 0$;
4     $a^* \leftarrow \emptyset$;
5   **foreach** $a \in A \setminus F$ **do**
6         $V_a \leftarrow updateVehicles(V, a)$;
7         $IR_a \leftarrow simulate(L, V_a, C)$;
8   **if** $IR_a > IR_p$ **then**
9             $IR_p \leftarrow IR_a$;
10             $a^* \leftarrow a$;
11     $V \leftarrow updateVehicles(V, a^*)$;
12     $F \leftarrow updateTabu(F, a^*)$;
13   **if** $IR_p > IR^*$ **then**
14         $IR^* \leftarrow IR_p$;
15         $V^* \leftarrow V$;
16         $B \leftarrow B \cup a^*$;
17 return B, V^*, IR^*;

best solution V^* and its associated index of regularity IR^* are stored during the optimization process (lines 14 - 16) and returned as output of Step 3 once the last iteration ends (line 17).

Step 4: automated evaluation. In the forth step of the delay management algorithm the quality of result obtained in the optimization step is automatically analyzed. In particular given IR^*, and IR_0 the optimized solution V^* is proposed to the operator in Step 5 only if $IR^* - IR_0 \geq S$ where S is a quality threshold. Otherwise the procedure starts over from Step 1 reading a new current solution.

Step 5: operator's intervention. In this step the solution V^*, proposed by the procedure, is submitted to the operator. The operator can either accept the solution, possibly making changes to V^* if needed, or reject it. In case of rejection the algorithm restart from Step 1, otherwise it moves to Steps 6.

Step 6: crew re-scheduling. The crew scheduling is re-optimized taking into account the new vehicle scheduling V^*. The output of this step is a new crew scheduling C^* that together with V^* will be the input for Step 7. In section 3.2 we present an algorithm to deal with the crew re-scheduling problem in a real time environment.

Step 7: dispatching a new course-of-actions. In this step the new schedules V^* and C^* are implemented generating actual modifications of the vehicles and crews scheduling.

The system operates, in a continuous optimization fashion, after Step 7 restarts from Step 1 reading the new current situation that was possibly modified by the previous round of optimization.

3.2. Crew Re-Scheduling

As mentioned before, once the vehicle scheduling has been modified, both due to delays and as a result of possible actions, crew scheduling has to be adapted. One possible approach is to apply the algorithm used for planning the service, though it must be applied to a very limited subproblem, with respect to the original one. However the characteristics of the subproblem, in terms of size, objectives, and constraints suggest an ad hoc approach. Indeed the main objective is no longer to minimize the cost, but to minimize the changes with respect to the planned service, in order to simplify the application. Moreover the regulations applied in the planning phase can be partially relaxed. The other important aspect is that in the re-scheduling subproblem we have to account for the fact that drivers are already on duty and they expect to work in the same period as in the initial planning. Notice also that it may happen that, with the new vehicle scheduling, some duties remain uncovered, not being possible to involve other personnel. This implies that, though in a limited extent, the re-scheduling problem has to consider also changes in the vehicle scheduling, eliminating parts of service that are not possible to cover, limiting the inconvenience on the users. In some way, the re-scheduling problem is tackling a joint vehicle scheduling - crew scheduling - rostering problem.

The problem can be described as follows. We are given a set of (re-scheduled) vehicle duties, and a set of drivers. Each vehicle duty may be split into pieces of work to be assigned to drivers. Each piece of work has a starting time and an ending time (possibly recomputed with respect to the planned one), and a starting and ending place. A penalty cost is associated to each piece of work, giving the cost of skipping it. Note that the vehicle duties can be split in multiple ways and a set of alternatives must be considered. For each driver we are given the remaining maximum working time, and the maximum number of remaining pieces of work he/she can be assigned to. In addition, since the decisions are taken when the service is carried out, we also know the assignment of drivers to the currently worked pieces. The problem consists of finding an assignment of the pieces of work to the drivers, maintaining the assignment of the currently worked pieces, so that the (relaxed) regulations are met, the cost for possible extra allowances is minimized, and the service is covered, or alternatively, the amount (or the penalty cost) of uncovered service is minimized.

Since the problem is very complex and can potentially involve millions of variables a column generation approach seems to be quite natural (see Desaulniers et al. (2005) and Gualandi and Malucelli (2013)). Moreover, with this approach part of the complexity of the problem, due to the involved regulations, is transferred to the pricing subproblem which can be easily tailored to take into account additional features and requirements. We utilize a directed acyclic graph $G = (N, A)$ to support the modeling. The node set N is partitioned into a set of drivers N_d, including possible spare ones, and set of pieces of works N_p. The set of arcs denote possible compatibilities between pieces of works, that is there is an arc $(j, j') \in A$ if pieces of work j and j' can be assigned consecutively to the same driver. Moreover there are arcs (i, j) connecting each driver i to pieces of work that can be assigned to him/her, that is that are compatible with the regulations about breaks. Notice that, if one driver i is currently on duty there are only arcs (i, j) such that j is the current piece of work assigned to i or possible modifications (shortened or lengthened pieces of works derived from j). A path in G starting form a node i in N_d and satisfying certain additional constraints, corresponds to a feasible completion of duty for driver i.

Defined A_{hr} as the incidence matrix of the path r, that is $A_{hr} = 1$ if and only if path r covers the (atomic) piece of work h (which is a unsplitable portion of vehicle duty). The idea of the approach is to use two sets of 0-1 variables in a set partitioning framework. In particular, variable x_r associated to each feasible completion (i.e., feasible path) r and variable y_h associated to each piece of work $h \in N_p$ and equal 1 if the piece of work h is not covered. Let R denote the set of feasible completions, $R_d \subset R$ the subset of completions associated with driver $d \in N_d$, c_r the overall cost of a completion and c'_h the penalty cost for not covering atomic piece of service h. The optimization problem is then:

$$\min \sum_{r\in R} c_r x_r + \sum_{h\in N_p} c'_h y_h \tag{3}$$

$$\sum_{r\in R} A_{hp} x_r + y_h = 1 \quad \forall h \in N_p \tag{4}$$

$$\sum_{r\in R_d} x_r \leq 1 \qquad \forall d \in N_d \tag{5}$$

$$x_p, y_h \in \{0, 1\} \qquad \forall r \in R, \forall h \in N_p \tag{6}$$

Notice that, since every path starts from a node of N_d, every completion is already assigned to a driver. The presence of spare drivers can be considered by adding suitable nodes to N_d. These problems can be approached by standard column generation methods where the model 3-6 is the master problem and in the pricing phase completions r are generated solving constrained shortest path problems (see Beasley and Christofides (1989)) on graph G. Notice that in the case of driver re-scheduling described in this paper, these paths contain 3-4 nodes at most, so they can be enumerated quite efficiently; for more complex cases algorithms as those described in Righini and Salani (2008) and Gualandi and Malucelli (2012) can be employed.

4. Experimental tests

In order to evaluate our approach we have implemented the *IR optimization algorithm* (Step 3) and the *crew re-scheduling algorithm* (Step 6) in *Python* (see Hart et al. (2012)) using *Pyomo 4.0* as MIP modeling library and *COIN-OR CBC 2.9* (see Lougee-Heimer (2003)) as MIP solver. All tests were done on an Intel Core i7-4702hq with 8 GB of RAM using Linux Mint Debian Edition (64 bits).

We tested both algorithms on 60 different real scenarios taken from a single bus line of Azienda Trasporti Milanese (ATM) of Milan. They refer to bus line 92 during the third week of May 2014 from 07:00 to 20:00. The test refer to a single line since the resources of that line are not shared with other lines, the trips are relatively frequent and the issues emerged in the past where sufficiently challenging to test the whole delay management system. In a preliminary analysis of historical data we found that there is no correlation between travel times and the gap between vehicles. This suggests that, in the ATM case, the effect of dwell times is not significant. Thus we decided to neglect this component in the simulator.

Following the results obtained by ATM in a preliminary validation phase (see Malagoli and Marzorati (2015)), the *IR optimization algorithm* has been tested using the piece-wise penalty function (see section 2.2) with $\alpha = 0.002$, $\beta = 0.007$, $\gamma = 0.002$, $\delta = 0.4$, $\theta_1 = 120[s]$, $\theta_2 = 120[s] + 10\%$ of the expected headway and $\theta_3 = 120[s] + 30\%$ of the expected headway. For comparison purpose we take also into account the 0-1 ATM penalty function with $t = 180[s]$. In all scenarios n for the *IR Optimization Algorithm* is equal to 2 hours, f is equal to 3, 10 rounds of simulation are considered and a maximum number of 10 improving actions is imposed.

In the crew re-scheduling phase a simplified regulation is adopted: each driver shift has a tolerance of three minutes in the starting time and in the breaks and can include up to 20 minutes of extra work. Moreover, in the objective function (3) c'_h is equal to twice the length in seconds of the piece of service $h \in N_p$ not covered and $c_r = \psi_r + \epsilon_r$ where ψ_r is equal to the extra time, in seconds, worked in the completion $r \in R$ while ϵ_r is equal to 1 if the pieces of service of the completion $r \in R$ are assigned to a different driver with respect to the planned one and 0 otherwise.

In table 1 results of the experimentation are reported. The table is spit into hours of the day and includes the following columns: *In.* is the scenario id, IR_0 is the unmanaged index of regularity (*Step 2*), IR^* is the best value of the index of regularity found (*Step 3*), 01_0 is the unmanaged index of regularity computed with the ATM 0-1 penalty function 01^* is the value of index of regularity computed with the ATM 0-1 penalty function on the best solution (corresponding to solution IR^*), *IR%* represents the percentage difference between IR^* and IR_0, 01% is the percentage difference between 01^* and 01_0, Ac. reports the number of suggested actions to improve the regularity, *Unc.* represents the amount of trip time (hh.mm.ss) that cannot be covered if all improving actions proposed are implemented without crew re-scheduling, *Unc** is the trip time not covered after crew re-scheduling, *Extra* extra work required by the crew re-scheduling, *Sw.* number of pieces of service that are assigned to a different driver with respect to the planned one, *Time* execution time in seconds of the crew re-scheduling algorithm. In the last row of the table (*AVG*) average values are reported.

Analyzing the values in the table it is clear that the *IR optimization algorithm* is able to find, in less then 5 minutes, at least one improving solution in all scenarios but one (scenario 21). The extent of the improvement has high variability, it is usually about 4% and can go up to almost 30%. We observed that it is usually higher in the first hours of the day when more improving actions are available. This results is obtained with, on average, less than 4 improving actions and it is worth noting that there is not clear correlation between the number of actions proposed and the improvement in the quality of the solution. The drawbacks of the 0-1 penalty function pointed out in section 2.1 are confirmed by our experimental analysis. Indeed the value of 01^* is lower than the value of 01_0 in almost half of the scenarios meaning that improving solutions would be often discarded if the 0-1 penalty function is used.

The crew re-scheduling algorithm is always able to sensibly reduce the uncovered pieces of work. In particular on average the service time not covered decreases from more than 1 hour to about 6 minutes and in almost 90% of the scenarios with the new crew scheduling all pieces of work are covered. It is worth noting that this results is obtained with little extra work (less than eight minutes) and that a piece of work is only occasionally assigned to a different driver in the new scheduling. The algorithm runs in, on average, about 30 seconds that are almost equally split between the pricing phase and the solution of the master problem in the column generation procedure.

Table 1. Experimental Results

In.	IR_0	IR^*	IR%	01_0	01^*	01%	Ac.	Unc.	Unc*	Extra	Sw.	Time
From 7:00 to 10:00 - Average Expected Headway: 5 m												
1	63.48	70.47	11.01	88.67	92.05	3.81	5	02.05.12	00.00.00	00.06.11	0	58.63
2	75.95	81.36	7.12	93.89	93.84	-0.05	6	04.31.12	00.00.00	00.11.33	0	142.38
3	74.05	77.49	4.65	92.68	91.92	-0.82	6	04.59.17	00.21.43	00.14.04	9	192.30
4	71.39	73.92	3.54	90.57	90.22	-0.39	3	03.20.14	02.24.53	00.00.44	9	53.75
5	71.12	73.89	3.89	89.18	89.25	0.08	2	01.56.00	00.10.00	00.00.05	0	14.39
6	68.26	70.21	2.86	87.47	87.40	-0.08	4	03.23.24	00.00.00	00.15.41	0	128.05
7	50.10	62.96	25.67	80.60	87.71	8.82	7	00.02.40	00.00.00	00.11.07	0	140.01
8	27.71	35.73	28.94	86.21	91.05	5.61	6	03.35.19	01.34.18	00.09.55	0	119.08
9	75.72	78.26	3.35	93.89	94.50	0.65	6	00.00.00	00.00.00	00.09.38	0	115.32
10	82.31	85.79	4.23	94.90	94.42	-0.51	4	00.00.00	00.00.00	00.08.35	0	45.76
11	69.54	74.56	7.22	88.94	88.32	-0.70	8	00.18.49	00.00.00	00.08.50	0	230.08
12	69.71	72.52	4.03	88.47	88.43	-0.05	6	01.29.48	00.10.10	00.01.45	3	141.93
13	65.95	67.82	2.84	86.51	86.47	-0.05	3	03.20.33	00.00.00	00.06.02	0	23.32
14	64.17	64.89	1.12	85.65	85.75	0.12	1	05.28.28	00.18.23	00.03.00	2	0.75
15	26.41	31.94	20.94	88.63	91.54	3.28	9	02.36.57	00.00.00	00.12.35	0	208.30
16	35.78	40.04	11.91	92.70	94.81	2.28	6	03.35.36	01.37.53	00.06.08	1	114.55
From 10:00 to 14:00 - Average Expected Headway: 8 m												
17	70.43	70.80	0.53	88.09	88.09	0.00	1	00.00.00	00.00.00	00.03.00	0	10.65
18	68.82	69.34	0.76	86.01	86.27	0.30	1	00.00.00	00.00.00	00.03.00	0	0.82
19	67.00	68.13	1.69	84.90	85.50	0.71	1	00.00.00	00.00.00	00.03.00	0	1.62
20	66.82	66.89	0.10	85.14	85.14	0.00	1	00.00.00	00.00.00	00.03.00	0	0.63
21	63.56	63.56	0.00	83.44	83.44	0.00	0	00.00.00	00.00.00	00.00.00	0	0.11
22	62.55	62.77	0.35	82.22	82.42	0.24	1	00.00.00	00.00.00	00.00.00	0	0.53
23	63.89	64.07	0.28	82.90	82.90	0.00	1	00.00.00	00.00.00	00.00.00	0	0.51
24	64.23	64.31	0.12	83.18	83.18	0.00	1	00.00.00	00.00.00	00.00.00	0	0.87
25	65.73	65.77	0.06	86.51	86.51	0.00	1	00.00.00	00.00.00	00.03.00	0	0.61
26	63.09	63.98	1.41	84.61	84.89	0.33	2	00.00.00	00.00.00	00.08.00	0	6.47
27	62.32	63.66	2.15	83.81	84.37	0.67	1	00.00.00	00.00.00	00.03.00	0	0.58
28	61.68	62.52	1.36	83.44	83.82	0.46	1	06.17.03	00.00.00	00.00.35	0	0.62
29	62.56	62.62	0.10	83.66	83.66	0.00	1	00.00.00	00.00.00	00.04.00	0	0.58
30	61.82	62.13	0.50	83.21	83.25	0.05	1	00.00.00	00.00.00	00.01.00	0	0.86
31	61.06	61.46	0.66	82.99	82.99	0.00	1	00.00.00	00.00.00	00.00.00	0	0.50
32	59.04	60.56	2.57	82.29	82.43	0.17	3	02.54.38	00.00.00	00.11.20	0	6.07
From 14:00 to 17:00 - Average Expected Headway: 7 m												
33	62.12	62.47	0.56	81.84	81.84	0.00	1	00.00.00	00.00.00	00.00.00	0	0.52
34	62.25	62.69	0.71	82.19	82.49	0.37	5	00.00.00	00.00.00	00.07.20	0	7.07
35	52.32	53.00	1.30	81.24	81.47	0.28	3	00.00.00	00.00.00	00.12.00	0	4.94
36	51.06	51.51	0.88	80.72	80.76	0.05	4	00.00.00	00.00.00	00.10.00	0	2.03
37	49.32	49.98	1.34	81.23	81.52	0.36	2	00.00.00	00.00.00	00.11.00	0	1.21
38	49.72	51.04	2.65	81.47	82.09	0.76	4	00.00.00	00.04.59	00.09.00	0	0.81
39	58.62	59.51	1.52	81.44	81.44	0.00	2	00.00.00	00.00.00	00.08.24	0	2.83
40	53.38	55.50	3.97	80.95	81.82	1.07	5	05.20.43	00.00.00	00.15.38	0	11.49
41	54.61	55.78	2.14	82.02	82.21	0.23	3	01.12.28	00.00.00	00.18.21	1	18.80
42	52.20	52.87	1.28	80.50	80.86	0.45	3	02.12.28	00.00.00	00.09.00	0	1.37
43	51.68	61.20	18.42	82.08	82.25	0.21	1	03.14.16	00.00.00	00.03.00	0	0.63
44	73.40	82.16	11.93	82.33	82.42	0.11	1	00.00.00	00.00.00	00.01.37	0	0.74
From 17:00 to 20:00 - Average Expected Headway: 6 m												
45	56.25	58.52	4.04	79.15	80.32	1.48	6	00.00.00	00.00.00	00.13.00	0	0.95
46	55.67	57.91	4.02	79.64	80.66	1.28	6	00.00.00	00.00.00	00.07.47	0	0.88
47	56.59	59.54	5.21	78.38	79.50	1.43	6	00.00.00	00.00.00	00.09.47	0	0.86
48	57.19	58.87	2.94	78.47	78.03	-0.56	7	00.00.00	00.00.00	00.21.21	0	1.48
49	56.51	57.63	1.98	78.11	76.30	-2.32	6	00.22.17	00.00.00	00.19.48	2	1.02
50	56.04	57.04	1.78	77.89	74.88	-3.86	4	00.00.00	00.00.00	00.06.59	0	0.87
51	54.87	54.91	0.07	78.14	72.84	-6.78	1	00.00.00	00.00.00	00.00.00	0	2.23
52	53.84	54.98	2.12	77.95	75.21	-3.52	2	00.00.00	00.00.00	00.00.00	0	1.25
53	71.01	71.35	0.48	81.71	81.74	0.04	3	00.00.00	00.00.00	00.03.00	0	0.61
54	41.42	42.55	2.73	81.40	81.23	-0.21	3	00.00.00	00.00.00	00.04.45	0	0.60
55	55.30	57.58	4.12	78.78	79.12	0.43	8	03.04.58	00.00.00	00.13.43	0	1.17
56	55.21	55.66	0.82	79.22	78.18	-1.31	8	01.43.12	00.00.00	00.20.17	0	1.41
57	53.03	53.89	1.62	77.17	76.21	-1.24	7	03.52.07	00.00.00	00.17.46	0	0.91
58	51.16	51.91	1.47	75.74	73.13	-3.45	10	01.18.21	00.00.00	00.21.35	0	0.91
59	49.02	49.25	0.47	75.51	72.44	-4.07	2	00.00.00	00.00.00	00.01.30	0	1.35
60	48.04	48.93	1.85	73.88	70.17	-5.02	4	00.00.00	00.00.00	00.07.53	0	3.55
AVG	59.30	61.38	3.97	83.41	83.46	0.02	3.62	01.12.16	00.06.42	00.07.23	0.45	30.55

5. Conclusions

In this paper we have presented an online system for the real-time delay management in the context of public transportation. A prototype of the system has been implemented and tested on a set of real instances. The experimental results show the effectiveness of the procedures and demonstrate that this approach represents a solid starting point for future developments. An experimental phase on a real time environment has been started.

Further investigations will focus on three different areas to improve the quality of the solution found as well as the applicability of the approach. In particular, in our implementation, the estimation of traveling and inter-links times is based on historical data. The current situation of the traffic flow is explicitly taken into account only in the first phase of the simulation when the position and the status of vehicles are retrieved from the AVM. This, in case of severe disruptions, may lead to inconsistent scenarios in which the situation represented by the simulation is considerably different from the reality. However, taking into account the evolution of the traffic flow during the day could be evaluated for example by keeping track of the actual traveling times coming from AVM observations and dynamically modify the historical data used in the simulation.

Another area that can benefit from further experimentations is the crew re-scheduling algorithm. In fact, even if our model can already handle the introduction of spare drivers, we were unable to perform any reliable test since no consistent data on the number and cost of spare drivers were provided.

Finally, we would like to investigate the applicability of our approach in multi-lines scenarios. We have already tested our algorithms on scenarios with up to four independent transportation lines. In this case our procedure scales up very well and with the adequate computational power it offers the same level of performance as in the single-line scenario. However, when transportation lines taken into account are not independent and share drivers, relief points and part of the paths then modifications to our approach may be required to provide high quality solutions that take advantage of the interdependence of the lines.

References

Barabino, B., Di Francesco, M., Mozzoni, S., 2013. Regularity diagnosis by automatic vehicle location raw data. Public Transport 4, 187–208.

Bartholdi-III, J.J., Eisenstein, D.D., 2012. A self-coordinating bus route to resist bus bunching. Transportation Research Part B 46, 481–491.

Beasley, J.E., Christofides, N., 1989. An algorithm for the resource constrained shortest path problem. Networks 19, 379–394.

Berger, A., Hoffmann, R., Lorenz, U., Stiller, S., 2011. Online railway delay management: Hardness, simulation and computation. Simulation 87, 616–629.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J., 2014. An overview of recovery models and algorithms for real-time railway rescheduling. Transportation Research Part B: Methodological 63, 15–37.

Clausen, J., Larsen, A., Larsen, J., Rezanova, N.J., 2010. Disruption management in the airline industryconcepts, models and methods. Computers & Operations Research 37, 809–821.

Desaulniers, G., Desrosiers, J., Solomon, M. (Eds.), 2005. Column Generation. Springer US. doi:10.1007/0-387-25486-2_2.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers.

Gualandi, S., Malucelli, F., 2012. Resource constrained shortest paths with a super additive objective function, in: Principles and Practice of Constraint Programming. Springer Berlin Heidelberg, pp. 299–315.

Gualandi, S., Malucelli, F., 2013. Constraint programming-based column generation. Annals of Operations Research , 1–22.

Gualandi, S., Malucelli, F., Tresoldi, E., 2014. Disruption Management in Public Transport: Mathematical Analysis of Service Regularity Issues. Technical Report. Politecnico di Milano.

Hart, W.E., Watson, J.P., Woodruff, D.L., 2012. Pyomo Optimization Modeling in Python. Springer.

Hickman, M.D., 2001. An analytic stochastic model for the transit vehicle holding problem. Transportation Science 35, 215–237.

Jespersen-Groth, J., Potthoff, D., Clausen, J., Huisman, D., Kroon, L., Maróti, G., Nielsen, M.N., 2009. Disruption management in passenger railway transportation. Springer.

Li, J.Q., Mirchandani, P.B., Borenstein, D., 2009. A lagrangian heuristic for the real-time vehicle rescheduling problem. Transportation Research Part E: Logistics and Transportation Review 45, 419–433.

Lougee-Heimer, R., 2003. The common optimization interface for operations research: Promoting open-source software in the operations research community. IBM Journal of Research and Development 47, 57–66.

Malagoli, L., Marzorati, M., 2015. Disruption - Analisi funzione obiettivo. Technical Report. ATM. Internal report in Italian.

Righini, G., Salani, M., 2008. New dynamic programming algorithms for the resource-constrained elementary shortest path problem. Networks 51, 155–170.