AUTOMATED, PASSENGER TIME OPTIMAL, ROBUST TIMETABLING, USING INTEGER PROGRAMMING

P. H. A. Sels [1,2,3], T. Dewilde [1] , P. Vansteenwegen [4] and D. Cattrysse [1]
[1] Center for Industrial Management,
Katholieke Universiteit Leuven, Belgium.
[2] R&D, Logically Yours BVBA, Antwerp, Belgium. Email:sels.peter@gmail.com
[3] Department of Network Access,
Infrabel, The Belgian Railway Infrastructure Company, Belgium
[4] Department of Industrial Management,
Ghent University, Ghent, Belgium.

ABSTRACT

To design an optimal passenger train timetable one should choose a quality criterium or a combination of criteria. We consider the main quality criterium from a passenger perspective: journey time. This means that the expected time all passengers will spend when our timetable is put in practice is minimal, even taking into account typical train delays. From a train operator or rail infrastructure management company perspective, there are further concerns too, like the number of train units that has to take part in this schedule, their frequency, the number of drivers and other crew members. These factors are all related to cost to maintain the schedule but are here considered secondary and indeed, are here kept constant.
We consider only the passenger criterium here. We analytically derive total stochastic expected passenger time as a closed formula, linearize it and use it as a goal function for optimizing the schedule using a mixed integer linear programming model.
We applied this to all 224 current Belgian train relations, passing 550 train stations and calculated an optimal schedule in 3 hours. We believe this mathematically optimal approach is unique, in its detailed model of expected, stochastic passenger time, in its scale of implementation and in its use of actual current data from practice.

KEYWORDS

timetabling, quality of service of passenger trains, expected passenger time, stochastic optimisation, mixed integer linear programming

RELATED RESEARCH OVERVIEW AND COMPARISON

Concerning linear modeling of a train service, a lot of theoretical research has been carried out already. (Serafini, Ukovich, Schrijver, Kroon, Nachtigall, Goverde, Liebchen). The first schedule that has been mathematically calculated, be it only for feasibility and not for optimality, was the one for the Netherlands, and is due to Kroon *et al*. (Kroon *et al.* 2007). Whether this was the schedule actually put in practice is unclear. Christian Liebchen claims to have produced the first schedule that was mathematically calculated, both for feasibility and including some partial optimization criteria (Liebchen2007). This was the schedule for the Berlin Underground, containing 37 train relations. We go further here, by firstly handling all passenger trains in a whole country, Belgium, and also using a more complete optimisation goal function: passenger time, which also includes a degree of robustness in a natural and consistent way. The idea of using expected passenger time as a goal function to minimize is based on research from Vansteenwegen et al. (Vansteenwegen 2006&2007, Dewilde 2011). The great power of this goal function is that it gives an automatic and sensible trade-off between a speedy yet robust service, where buffers are neither too large nor too small respectively.

INPUT DATA

We want to minimise the total expected passenger time. For this, we will weigh the time by passenger numbers actually present on each train. Also, the time expected is predicted from the time each section took in the past. This means that systematic delays present in the past, are expected to continue occurring in the future and this is automatically compensated for by inserting time buffers against these delays in the produced schedule. So passenger numbers as well as known historical delay distributions are input data to our procedure.

Passenger Numbers

Passenger numbers, on each section of each train, can be derived from passenger counts. However, these are often incomplete. A better way to obtain these numbers, is by deriving them from train subscription and ticket sales. We refer to (Sels 2011) for a procedure to obtain these passenger numbers.

Train Delays

Train delays can be derived from measurements which are usually carried out by the railway infrastructure company. Common practice is that a time measurement is done for each train entering a station or other time table point. Also at exit from this station or time table point, a new time measurement is done. By subtracting both one can derive the local delay in practice, which is to be considered on top of the necessary minimum time. We assume a negative exponential probability distribution for this delay distribution. This allows us to further analytically derive expected passenger time. In fact, these delays with their distributions, are the only stochastic component and the sole reason that expected passenger time becomes stochastic.

In the selection of the historical delays, it is important that only delays that are considered to be reproduced in the future too, are contained in the distribution. For sure, systematic delays due to infrastructure problems are to be considered here. A single delay attributed to a malfunctioning train unit should be discarded from the distribution. Non-systematic delays due to infrastructure, like a upper line problem caused by accidental lightning should also be discarded. It sure is a lot of work to separate the systematic, infrastructure related from the accidental, non recurring delays. We will produce a schedule which is robust against these systematic delays. We believe that this approach will correctly and carefully weigh problem areas versus non-problem areas and is preferable. When systematic delay data is not immediately available, an easier, be it less optimal way, to get a reasonably robust schedule is to consider delays that are averaged over all locations. We initially use this approach, where we impose an expected average delay of 6 seconds at each ride segment, one of 12 seconds for each dwell action, and one of 60 seconds for each transfer action.

TWO PHASED SOLUTION PROCEDURE

We now set up a procedure, in multiple steps, to derive passenger numbers on each section of each train. The procedure of letting passengers choose their train or trains, including transfers between them, from the different trains available , considering the current timetable available, we call *reflowing*. This is done using the modified Dijkstra algorithm (Sels 2010). We there also describe how a more neutral initial schedule than the current one can be used to get rid of any bias towards specific non-optimal transfer times, that are already, inherently present in the current timetable.

Next, we need to decide on a new, more optimal timing for the schedule. We call this procedure *retiming*. Retiming is a much harder problem. Contrary to reflowing, where each passenger can just choose his route on his own, retiming cannot be as easily decoupled. A large integer linear programming model is set up. This model contains all necessary coupled constraints. One example is *time continuity* where a begin of each action in the action graph is equalled to the end of all predecessor actions. This holds for all types of actions: ride, dwell and transfer. The biggest number of constraints are the *headway constraints*, where for each pair of trains on a certain resource, a separation time of 3 minutes is enforced. For speeding up the MILP solver used, a certain type and number of *cycle constraints* is also enforced. Also, the relative amount of time supplements allowed per train line is limited to maximum 25%.

Reflowing takes as input, a given schedule, meaning the time t_e for each action edge e, being a ride action, dwell action or transfer action, and produces flow numbers f_e, representing the number of passengers on these same action edges. Retiming, then, takes these flow numbers f_e per edge and produces a duration d_e per edge. The result of these two phases is that each action edge has a flow f_e and a duration d_e. This is represented in figure 1. Of course, when a schedule is *retimed*, passengers can react differently on this new schedule by choosing another route, for example, because a transfer is becoming too tight or too long. The first case makes the risk of missing the transfer too high. The second case means the certainty of having to wait very long.

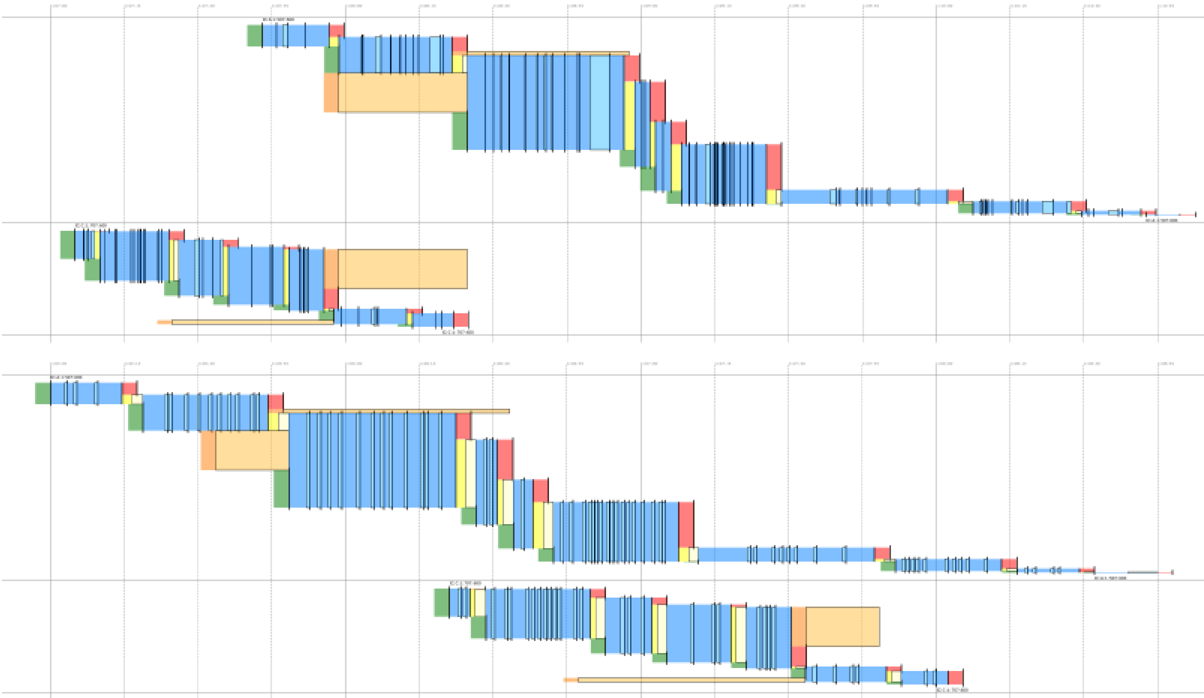
ITERATION UNTIL CONVERGENCE

We can consider the subsequent *reflowing* and *retiming* an interplay of the parties: passenger and rail companies. Both processes adapt to the change of the preceding process. This means *iteration* should be carried out over these 2 subsequent phases, until convergence, before putting any schedule in practice. Indeed, reflowing should be done in computer memory, not by actual passengers in practice, on a still sub-optimally timed schedule.

RESULTS AND DISCUSSION

We produce both flow numbers f_e and durations d_e, and also actual times for each time table point. We defined a new way to display these results in two dimensions. The vertical dimension represents the f_e numbers. The horizontal dimension represents the durations d_e of subsequent actions and also clearly indicates their order in time. Figure 2 gives a representation of two trains, on the top for an unoptimized, current schedule and at the bottom for an optimized schedule. This is the result of one *reflow-retime* iteration only.

Figure 1: Current schedule of two trains (top) and optimized schedule of same two trains (bottom)



For proving that a schedule is better than the one we started with, it is important to be able to show that the expected passenger time has been lowered. This is shown in figure 2. The left half of this figure shows *train time* while the right half shows *passenger time.* In each graph, all times are categorised as summed ride, dwell, transfer times, each time also as minimum part (darker colour below) and supplement part (lighter colour above). We also added the headway times and their expected times corresponding with headway supplements on top. The blue shaded areas refer to the fact that for these actions, the preceding ride action and the action itself have been convoluted to calculate the expected time.

CONCLUSIONS

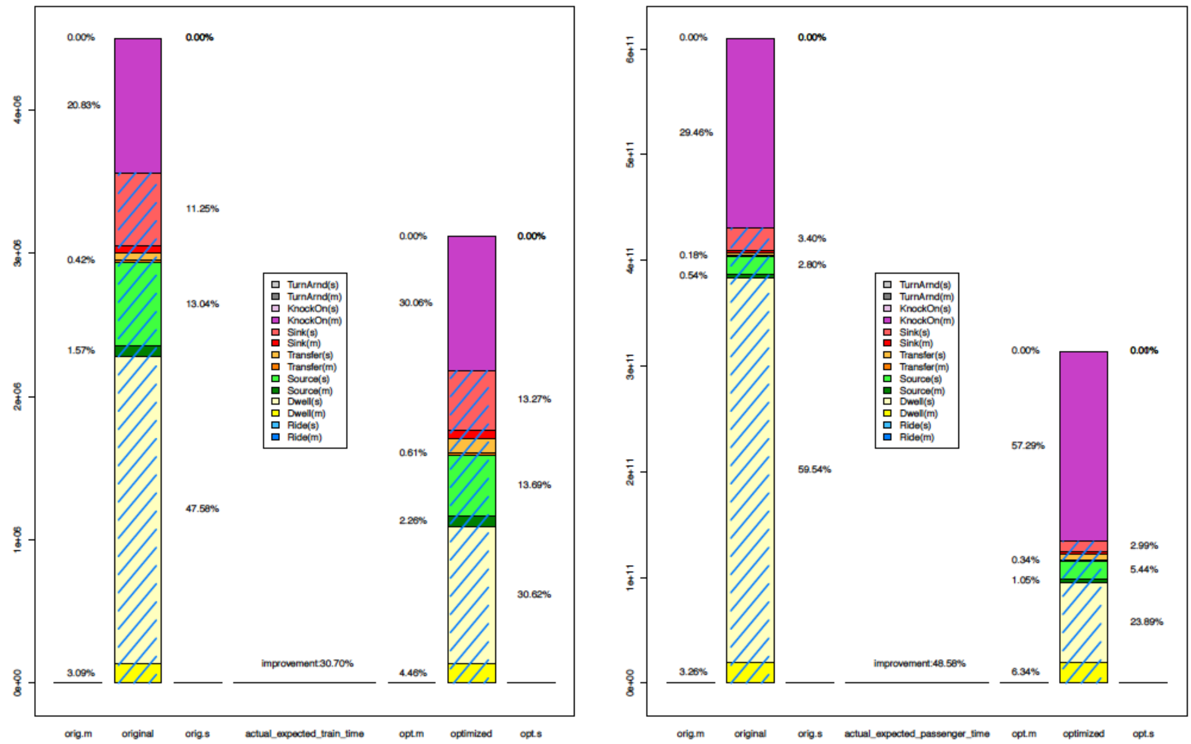
We constructed a two phase procedure that models the passengers decisions (reflowing) and the railway companies decisions (retiming). This can be iterated over in computer memory before the final optimal schedule is put into practice. After convergence, the produced schedule has the lowest possible expected passenger time, taking into account typical delays and passenger flows on every section. One iteration, for all hourly passenger trains in Belgium, currently takes about 3 hours calculation time.

FURTHER RESEARCH

First, for ride dwell sequences, we need to further tune the goal function. Dwell action minima that have a probability to be exceeded, in the cases that they are exceeded, are still overly penalised, resulting in a schedule that weighs robustness a little too high resulting in a lower speed of service.

Secondly, we want to iterate over *reflowing* and *retiming* until convergence.

Figure 2: Results on 224 trains in Belgium:
Expected Train Time Reduction of 30% (left) and Expected Passenger Time Reduction of 48% (right).
Optimistic reduction results are due to currently still over-stressing of robustness of ride-dwell sequences.



ACKNOWLEDGMENTS

We thank Infrabel for the cooperation with Logically Yours BVBA and for supplying the necessary data.

REFERENCES

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., "Defining Robustness of a Railway Timetable", Proceedings of RailRome, 2011.
Goverde, R.M.P., "Improving Punctuality and Transfer Reliability by Railway Timetable Optimization", Proceedings of the 5th TRAIL Annual Congress, TRAIL Research School, Delft, 1999.
Kroon, L., Dekker, R., Vromans, M.J.C.M., "Cyclic Railway Timetabling: A stochastic Optimization Approach", Geraets, F., Kroon, L., Sch obel, A., Wagner, D., Zaroliagis, C.D. (eds), Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science, pp. 41-66, 2007.
Liebchen, C., "Periodic Timetable Optimization in Public Transport", Operations Research Proceedings, 2007, Volume 2006, Part II, 29-36.
Nachtigall, K., "Periodic network optimization with different arc frequencies", Discrete Applied Mathematics, Vol. 69, pp. 1-17, 1996.
Schrijver, A., Steenbeek, A., "Spoorwegdienstregelingontwikkeling (Timetable Construction)" Technical Report, CWI Center for Mathematics and Computer Science, Amsterdam, (in Dutch), 1993.
Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., "Deriving all Passenger Flows in a Railway Network from Ticket Sales Data", Proceedings of RailRome, 2011.
Serafini, P., Ukovich, W., "A Mathematical Model for Periodic Scheduling Problems", SIAM Journal on Discrete Mathematics, Vol. 2, pp. 550-581, 1989.
Vansteenwegen, P., Van Oudheusden, D., "Developing railway timetables which guarantee a better service", EJOR, Vol. 173, pp. 337-350, 2006.
Vansteenwegen, P., Van Oudheusden, D., "Decreasing the passenger waiting time for an intercity rail network", Transportation Research Part B: Methodological, Vol. 41, pp. 478-492, 2007.