A recoverable robust solution approach for real-time railway crew rescheduling

Lucas P. Veelenturf[1], Daniel Potthoff[2], Dennis Huisman[3,4],
Leo G. Kroon[1,3], Gábor Maróti[3] and Albert P.M. Wagelmans[4]

[1] Rotterdam School of Management and ECOPT, Erasmus University Rotterdam
P.O.Box 1738, 3000 DR, Rotterdam, The Netherlands
Email: lveelenturf@rsm.nl

[2] Ab Ovo Germany

[3] Process quality and Innovation, Netherlands Railways

[4] Econometric Institute and ECOPT, Erasmus University Rotterdam

1 Introduction

In this paper we study real-time railway disruption management and we focus on crew rescheduling under uncertainty. Every day railway operations have to deal with disturbances and disruptions in their operations. Small-scale disturbances, often caused by minor delays, do not require a significant change of the schedules other than letting propagate (and hopefully diminish) the delays. This research however focuses on large-scale disruptions, often caused by failures of the infrastructure (e.g., broken overhead lines or defect signals), weather conditions or the unavailability of resources (e.g., defect rolling stock or delayed crew members). Such disruptions make the planned timetable, rolling stock and crew schedules infeasible, which inevitably requires a major adjustment of the schedules. Mind that the exact duration of the disruption is generally not known upon its start. For example, recovery works on a broken railway line segment may take 2 hours in an optimistic scenario, however they may stretch up to 4 hours in a pessimistic scenario.

Effective disruption management is key to a good operational performance of a train operating company. We refer to Jespersen-Groth et al. (2009) for a detailed description of the disruption management process. In the ideal situation the rescheduling of the timetable, rolling stock and crew schedules is integrated into one approach. However, this leads to problems of unsolvable complexity, and therefore most research focuses on rescheduling the resources sequentially: first the timetable, then the rolling stock, and the crew as last, even if this leads to some sub-optimality.

Current approaches to reschedule the crew use a deterministic approach to deal with the duration of the disruption: based on an initial estimation of the duration, the rescheduling step is carried out again and again when new information about the duration of the disruption becomes gradually available.

In this paper we consider methods that take the uncertainty in the duration of the disruption explicitly into account such that, if the disruption takes longer than expected, less rescheduling effort is necessary, and less trains are cancelled due to lack of crew. We assume that the timetable and rolling stock schedule have already been adapted based on the expected duration of the disruption. The primary criterion in assessing the quality of a schedule is to count the number of trains which cannot be covered by any crew. If trains have to be cancelled due to lack of crew, the rolling stock must also be rescheduled again. Other costs like operational and process costs are also taken into account, but have less priority.

We want to emphasize that our focus lies both on developing new methods and on practical applications. Our computational tests are based on railway crew scheduling instances that are quite challenging, even without the added uncertainty.

Algorithmic frameworks for dealing with uncertainty include the classical approaches of *robust optimization* and *stochastic programming*. These existing methods are not well-suited for real-time railway crew rescheduling problems with uncertainty due to their rigidity.

In this paper we propose a *quasi-robust optimization approach*, which is built upon the recently introduced concept of *recoverable robustness*. The main idea is to compute a good schedule for an optimistic duration of the disruption in such a way that it can easily be turned into a feasible schedule if the disruption takes longer than expected in the optimistic scenario. This is achieved by requiring that the computed crew duties have an alternative for trains for which it is uncertain that they will be operated.

Quasi-robustness is rather easy to incorporate into existing crew scheduling algorithms without substantially raising their running time. Furthermore, the approach admits to balance the robustness and the operational costs by requiring a certain percentage of the rescheduled duties to be recoverable robust.

We demonstrate the value of our approach by computational tests carried out on large-scale instances of Netherlands Railways (NS), the main operator of passenger trains in the Netherlands.

The contributions of this paper are summarized as follows.

- We describe disruption management methods for railway crew rescheduling under uncertainty.
- We develop a framework for dealing with the uncertainty about the duration of the disruption.
- We evaluate the proposed approach on large-scale crew rescheduling instances of NS.

This paper is organized as follows. We provide a brief overview of the literature in Section 2. In Section 3, we explain the quasi-robust solution approach for crew rescheduling problems. Section 4 describes that the method works well on real-life instances of NS. This paper is concluded in Section 5.

2 Literature overview

Huisman (2007), Rezanova and Ryan (2010) and Potthoff et al. (2010) study the short-term railway crew rescheduling problem. A common property of these models is that the disruption is considered to be deterministic, i.e. its duration is assumed to be known at its start.

However, uncertainty is often an essential part of the problem. Earlier research has developed two classical frameworks to explicitly deal with uncertainty. *Robust optimization* (Bertsimas and Sim (2003) and Ben-Tal and Nemirovski (2002)) and two-stage *stochastic programming* (Birge and Louveaux (1997) and Kall and Wallace (1994)).

Both robust optimization and stochastic programming lead to significantly more complex optimization problems than the underlying deterministic problem; usually realistic instances cannot be solved in (near) real-time planning. In addition, robust optimization is very conservative, while stochastic programming needs information about a probability distribution function, which is usually not available in practice.

Liebchen et al. (2007, 2009) introduced the concept of *recoverable robustness*. The aim of recoverable robustness is to overcome some shortcomings of classical robust optimization by considering a limited set of recovery actions. Our approach can be considered as an application of recoverable robustness to the crew rescheduling problem.

3 Quasi-robust optimization approach

3.1 Rescheduling under uncertainty

We assume that there is a limited set of scenarios. For each scenario s from the set of scenarios S, the set of tasks N_s is given. Here $\underline{s}$ is the scenario in S with the shortest duration of the disruption (the *optimistic* or *best case* scenario) and $\overline{s}$ is the scenario in S with the longest duration of the disruption (the *pessimistic* or *worst case* scenario). We assume the following relationship between all scenarios. Given the optimistic scenario $\underline{s}$, all other possible scenarios can be obtained by removing tasks from the reference scenario $\underline{s}$. The extra tasks that are cancelled due to a longer duration of the disruption in the pessimistic scenario $\overline{s}$ are *critical* tasks.

We refer to the start time of the disruption as time t_1. At time t_1 a minimum duration h_1 of the disruption is known ($\underline{s}$). Later, at time t_2 it becomes clear when the disruption ends (t_3). Often this time t_3 is later than the expected time $t_1 + h_1$. Usually this means that additional trains are cancelled such that some tasks cannot be executed.

In a wait-and-see approach, scenario $\underline{s}$ is used for the rescheduling at time t_1, which we will refer to as stage 1. At time t_2 the duties need to be rescheduled again if time t_3 is later than the expected time $t_1 + h_1$, which we refer to as stage 2.

Then the crew rescheduling problem under uncertainty can be stated as follows. Given an optimistic scenario $\underline{s}$, and a set of alternative scenarios $S \setminus \{\underline{s}\}$, find in stage 1 a new crew schedule valid for scenario $\underline{s}$, such that the sum of the cost of this schedule and the expected cost for the additional rescheduling in stage 2 is minimized. Computing the expected cost in stage 2 requires that the probability distribution of the scenarios is known. In general this is not known, therefore we only consider the cost of stage 2 of the scenario with the longest duration: $\overline{s}$.

3.2 Solution approach

We model the stage 1 problem of our quasi-robust optimization approach as a set covering problem with an additional constraint referring to the number of crew members which must get a recoverable robust duty. A duty obtained in stage 1 is called *recoverable robust* if the duty can also be performed by the crew member if in stage 2 the disruption appears to take longer than expected. For stage 2, the set covering problem does not contain this additional constraint, since then it is assumed that the duration of the disruption is known.

The solution approach for solving the set covering problem consists of a combination of Lagrangian relaxation and column generation which is based on Huisman et al. (2005) and Caprara et al. (1999). In typical applications, crew members can have a huge number of potential duties, therefore we employ a column generation approach where only prospective duties are considered. A column generation approach requires to solve the master problem multiple times. As our focus lies on real-time rescheduling with short computation times, we apply Lagrangian relaxation to reduce the complexity of the master problem.

4 Computational results

We tested our approach on instances of NS. For almost every instance, against some slightly higher cost for the optimistic scenario (without additional uncovered tasks), we can reduce the number of uncovered tasks in stage 2 if rescheduling was necessary since the pessimistic scenario took place. This is an important feature for the application in practice, since it means that against some slightly higher cost, we can prevent future cancellations of trains due to lack of crew if the disruption takes longer than expected. Moreover, our algorithm can explore the consequences of several robustness levels, and thereby help the decision makers to find the best balance between robustness and operational cost.

5 Concluding remarks

In this paper we study real-time railway crew rescheduling in case of large-scale disruptions. We propose a novel rescheduling approach that explicitly deals with the uncertain duration of the disruption.

We demonstrate the power of our approach on real-life crew rescheduling problems of NS. Our method is able to find solutions of reasonably good quality (proven by lower bounds) in a matter of minutes. The results show that quasi-robustness reflects the intuitive notion of robustness rather well.

Besides its methodological contributions, the method has good prospects to be valuable in practice. First, computations on challenging real-life instances reliably lead to good solutions. Second, the computation times of a few minutes are close to what is needed in real-time decision making during railway operations. Third, our approach is able to balance robustness requirements against operational and recovery costs. This allows the dispatchers to explore several solution variants and to make our algorithm reflect their insights.

References

Ben-Tal, A. and A. Nemirovski (2002). Robust optimization - methodology and applications. *Mathematical Programming 92*, 453–480.

Bertsimas, D. and M. Sim (2003). Robust discrete optimization and network flows. *Mathematical Programming 98*, 49–71.

Birge, J.R. and F. Louveaux (1997). *Introduction to Stochastic Programming*. New York: Springer.

Caprara, A., M. Fischetti, and P. Toth (1999). A Heuristic Method for the Set Covering Problem. *Operations Research 47*, 730–743.

Huisman, D. (2007). A column generation approach for the rail crew re-scheduling problem. *European Journal of Operational Research 180*, 163–173.

Huisman, D., R. Jans, M. Peters, and A.P.M. Wagelmans (2005). Combining Column Generation and Lagrangian Relaxation. In G. Desaulniers, J. Desrosiers, and M.M. Solomon (Eds.), *Column Generation*, GERAD 25th Anniversary Series, pp. 247–270. New York: Springer.

Jespersen-Groth, J., D. Potthoff, J. Clausen, D. Huisman, L.G. Kroon, G. Maróti, and M.N. Nielsen (2009). Disruption Management in Passenger Railway Transportation. In R.K. Ahuja, R.H. Möhring, and C.D. Zaroliagis (Eds.), *Robust and Online Large-Scale Optimization*, pp. 399–421. New York: Springer.

Kall, P. and S.W. Wallace (1994). *Stochastic Programming*. Chichester: John Wiley & Sons.

Liebchen, C., M.E. Lübbecke, R.H. Möhring, and S. Stiller (2007). Recoverable Robusness. Technical Report ARRIVAL-TR-0066, Algorithms for Robust and online Railway optimization: Improving the Validity and reliAbility of Large scale systems (ARRIVAL).

Liebchen, C., M.E. Lübbecke, R.H. Möhring, and S. Stiller (2009). The Concept of Recoverable Robustness, Linear Programming, and Railway Applications. In *Robust and Online Large-Scale Optimization*, pp. 1–27. New York: Springer.

Potthoff, D., D. Huisman, and G. Desaulniers (2010). Column Generation with Dynamic Duty Selection for Railway Crew Rescheduling. *Transportation Science 44*, 493–505.

Rezanova, N.J. and D.M. Ryan (2010). The train driver recovery problem - A set partitioning based model and solution method. *Computers & Operations Research 37*, 845–856.