Solving the Train Driver Recovery Problem

Extended Abstract

Natalia J. Rezanova
Informatics and Mathematical Modelling, Technical University of Denmark,
Richard Petersens Plads 1, Building 305, DK-2800 Kgs. Lyngby, Denmark,
phone: (+45) 45 25 33 89, fax: (+45) 45 93 12 35, e-mail: njr@imm.dtu.dk

David M. Ryan
The Department of Engineering Science, Faculty of Engineering,
The University of Auckland, Private Bag 92019, Auckland Mail Centre,
Auckland 1142, New Zealand, phone: (+64) 9 373 7599 ext. 88398,
fax: (+64) 9 373 7468, e-mail: d.ryan@auckland.ac.nz

The daily operations of the Danish railway operator DSB S-tog suffer from disruptions of various magnitude almost every day. Disruptions initiated by e.g. signalling problems or rolling stock failures cause train delays and cancellations.

Changes in the train schedule affect the train driver duties. If a train is cancelled or delayed, the driver assigned to the train task might not be able to reach the station of his next train departure in time. In practice, if a driver is not available for the train departure, another driver, for instance, a reserve, is assigned to the task. If there are no drivers available to cover the task on time, the train is delayed or cancelled, causing further disruptions.

The train driver recovery is performed by dispatchers, who often work under tremendous pressure. The size of the schedule (more than 2 000 train tasks, which are covered by approximately 270 drivers on a weekday) and a high frequency of the train departures with headways down to 2 minutes at certain network segments makes it difficult to find a good recovery solution fast. Drivers are in the worst case assigned to uncovered train tasks on the "first in first out" basis without considering the overall situation and, as a result, the train driver schedule is not *recovered*, i.e. the duties keep being modified from the originally scheduled duties for the rest of the day.

The railway crew disruption management has only received a very limited attention by OR practitioners. The crew re-scheduling problem for train driver duties disrupted due to the maintenance work on train tracks is solved by [1] for the largest passenger railway operator in The Netherlands. An integer programming approach to a simultaneous train timetable and crew roster recovery problem, tested on the New Zealand´s Wellington Metro line, is presented in [5].

The *Train Driver Recovery Problem* (TDRP) is aimed at finding the optimal set of feasible train driver recovery duties within a certain recovery period, such that all train tasks within the recovery period are covered, the number of modified duties within the recovery period is minimized, and the train driver schedule outside the recovery period is unchanged. In other words, the goal is to get back to the original schedule with as little change as possible.

Since disruptions might occur one after another during the day, the prototype for the Train Driver Recovery Decision Support System is based on solving TDRP instances sequentially. The changes caused by every disruption are applied to the original schedule as the disruption occurs, and the TDRP is resolved over a new set of duties, defined over a new recovery period. The term *original schedule* in this context stands for the schedule obtained by solving the previous TDRP instance, and *original duties* correspond to the optimal recovery duties obtained in the previous solution.

The recovery period has a rolling horizon, starting from the time when the disruption occurs and spanning over a certain time period, which is chosen to be sufficiently long to make a qualified recovery decision and sufficiently short to enforce the recovery to the original schedule as soon as possible.

The Train Driver Recovery Problem is formulated as a *set partitioning problem.* Let K be the set of train drivers to be involved in the recovery and N be the set of train tasks originally assigned to the involved drivers within the chosen recovery period. The set N can be empty, if, for instance, all drivers in K are reserve. Let P^k be the set of feasible recovery duties for driver $k \in K$ within the recovery period. A recovery duty is feasible, if it allows the driver to continue with his original duty after the end of the recovery period, if the recovery duty ends no later than originally scheduled, and if it contains all meal breaks assigned to the duty within the recovery period, each break starting no later than the originally scheduled break.

The cost c_p^k reflects the unattractiveness of the recovery duty $p \in P^k$, including the number of modifications from the original duty and other considerations. The objective function (1) of the model minimizes the total cost of the recovery solution. A binary decision variable x_p^k equals 1 if the duty $p \in P^k$ for the driver $k \in K$ is included in the solution and equals 0 otherwise. A binary parameter a_{ip}^k is used to define whether or not the task $i \in N$ is covered by the duty $p \in P^k$.

$$\text{(TDRP)} \qquad \min \sum_{k \in K} \sum_{p \in P^k} c_p^k x_p^k \tag{1}$$

subject to

$$\sum_{p \in P^k} x_p^k = 1 \qquad \forall k \in K \tag{2}$$

$$\sum_{k \in K} \sum_{p \in P^k} a_{ip}^k x_p^k = 1 \qquad \forall i \in N \tag{3}$$

$$x_p^k \in \{1, 0\} \quad \forall p \in P^k, \ \forall k \in K \tag{4}$$

The crew constraints (2) ensure that each driver is assigned to exactly one recovery duty in the schedule. The set partitioning constraints (3) ensure that each train task in the recovery schedule is covered exactly once.

The TDRP model (1) - (4) resembles the well-studied crew rostering model. As observed in [3], the linear programming (LP) relaxation of the model possesses strong integer properties, since every crew constraint (2) prohibits fractional solutions within the set of columns contributing to that row, i.e. forces the corresponding submatrix to be *perfect*. It means that if a fractional solution occurs, two or more drivers compete for one or more train tasks in their recovery duties. A few iterations of a Branch & Bound algorithm are then required to reach the optimal integer solution.

When a disruption occurs, only a small subset of drivers is directly affected. This set of drivers defines the initial set K to be involved in the recovery solution. Train tasks assigned to the initial set of drivers within the recovery period define the initial set of train tasks N.

Feasible recovery duties are constructed from directed graphs, formed individually for each driver $k \in K$. The source vertex in the graph represents the last task assigned to the driver prior the disruption. The sink vertex represents the first task assigned to the driver in his original duty after the end of the recovery period. The set of intermediate vertices in the graph represents all train tasks included in the recovery, which can potentially be covered by the driver. There is an arc between two vertices in the graph, if the two corresponding tasks could be covered in sequence by the driver. These arcs can represent train connections, deadheadings (riding as a passenger on a train) and meal break opportunities. A directed path through the graph for driver $k \in K$ from the source vertex to the sink vertex, containing at least one break opportunity arc for each originally scheduled meal break, represents a feasible recovery duty for the driver.

The LP relaxation of the TDRP is solved with a *column generation approach*. In order to keep the initial restricted master problem small, the first set of columns is collected using a *limited subsequence strategy*, described in [2]. Outgoing arcs of each vertex in the graph are sorted in the ascending order of their cost, which reflects the unattractiveness of the subsequent activity represented by the arc. The initial set of paths is collected, only allowing to expand each path with for example 2 or 3 most favourable subsequences at each vertex of the graph.

The pricing step of the column generation constructs a set of negative reduced cost columns. Dual values from the crew constraints (2) are applied on source vertices of driver graphs, while dual values from the set partitioning constraints (3) are applied on the train task vertices. A path, representing a feasible recovery duty with a negative cost, accumulated by adding the arc costs and subtracting the costs on vertices, represents a negative reduced cost column, which is a candidate to enter the basis of the restricted master problem.

The initial set of drivers K is often not enough to cover all trains in N due to the disruption. The LP relaxation of the TDRP is then infeasible. The solution space is then expanded by adding another driver to K, who might or might not already have train tasks assigned to him in his original duty. Expanding the solution space corresponds to adding constraints to the TDRP model. The strategy of choosing the driver to add to the set K is essential for the speed of reaching a feasible solution.

Preliminary strategies for the *constraint generation procedure* involve adding reserve and active drivers, who appear to be in the "neighborhood" of the disruption. The restricted master problem is resolved with new constraints, providing new dual values. The column generation procedure repeats. The constraint generation process continues until a feasible solution to the LP relaxation of the TDRP is found.

If any fractions occur in the solution, the Branch & Bound algorithm is applied. Since the fractions can only occur for variables, representing two or more drivers competing for the same train tasks, a *constraint branching* strategy initially proposed by [4] is appropriate and efficient to resolve the fractions.

DSB S-tog train driver schedule from year 2005 is used in order to test the prototype for the TDRP Decision Support System. Severe disruptions in form of cancelling between 25 and 75 train tasks at each disruption and using recovery periods between 2 and 6 hours at every run of the TDRP is applied. Preliminary test results show that each run of the TDRP takes between 0.2 and 20 seconds, including applying changes caused by the disruption to the schedule and reaching a feasible solution to the LP relaxation of the problem, solved with the column and constraint generation. All test instances produce integer solutions from linear programming relaxations, confirming the strong integer property of the problem.

References

[1] D. Huisman. A column generation approach to solve the crew re-scheduling problem. *Econometric Institute Report, Erasmus University of Rotterdam, The Netherlands*, (EI2005-54), 2005.

[2] D. M. Ryan. The solution of massive generalized set partitioning problems in aircrew rostering. *The Journal of the Operational Research Society*, 43(5):459–467, 1992.

[3] D. M. Ryan and J. C. Falkner. On the integer properties of scheduling set partitioning models. *European Journal of Operational Research*, 35(3):442–456, 1988.

[4] D M. Ryan and B. A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Aided Scheduling of Public Transport*, pages 269–280. North-Holland Publishing Company, 1981.

[5] C. G. Walker, J. N. Snowdon, and D. M. Ryan. Simultaneous disruption recovery of a trian timetable and crew roster in real time. *Computers & Operations Research*, 32(8):2077–2094, 2005.