A hybridised exact and local search method for robust train driver schedules planning

Ignacio Laplagne, Raymond S K Kwan, and Ann S K Kwan

School of Computing, University of Leeds, Leeds, LS2 9JT, UK
{ignacio, rsk, ann}@comp.leeds.ac.uk

When a train arrives at a station, it often stops for some time before continuing. The attention of a driver may be required throughout the train's stay at the station platform. Sometimes a train may be allowed to be unattended by immobilising it on arrival and mobilising it again before departure, but these activities take time and it might not be worth the trouble unless the stoppage at the station is significantly long. While the train is at a station, it may be an opportunity to relief the driver, i.e. letting the driver get off –either for a break or to go off-duty– and another driver come on the train to take over. It is usually preferred practice to relief drivers immediately on arrival of the train, which is also a convenient simplification from the scheduling point of view. However, as train delays are becoming prominent issues in the UK, utilising windows of relief opportunities at the stations may help improving the robustness of driver schedules. It may be argued that the mid-points of such time windows might be the most robust for changing drivers; nevertheless, operators will also have to consider schedule efficiency, and part of a relief opportunity time window might be critical for making efficient links. Hence, a comprehensive consideration of the full windows of relief opportunities is needed.

Generate and Select (GaS) is perhaps the most successful approach for automatic bus and train driver scheduling. During the generation phase, all the candidate shifts that satisfy the labour rules and parameters are constructed. Typically the number of candidate shifts may run into several hundred thousands. In the selection phase, the best minimal subset will be selected to form the schedule. In this approach, a relief opportunity is modelled by a (time, location) pair. Windows of relief opportunities could be modelled by a discrete set of contiguous (time, location) pairs, e.g. at each minute within the time window. However, this results in an exponential increase in the number of variables and/or restrictions in the model [1].

The multi-neighbourhood local search approach

In order to remove the simplification in the modelling of time windows, we devised a two-step, combined integer programming and local search approach. In the first step, windows of relief opportunities are approximated by their arrival time. The problem is then solved with GaS, using a branch-and-bound mechanism [3, 2]. After that, windows are restored and a second phase is carried out, in which a multi-neighbourhood local search tries to improve on the solution obtained in the first phase.

Apart from any moves that may be applicable to the non-windowed scenario, special consideration must be put in the way in which windows can be exploited. In our approach, a move is a composition of two steps. The first step is a specific perturbation to a subset of the shifts of the schedule. The second one is a generic procedure that, given a subset of relief opportunities R, analyses the possibility of altering the locations at which reliefs take place inside any windows that may belong to R. Each move selects this set R according to the characteristics of the perturbations it performs. For example, if a move splits a shift in two shifts, the second phase will probably receive a set that consists of the relief opportunities used at the ends of those two new shifts. Some of the moves we use have been originally developed by Shen and Kwan [4, 5] for a simplified version of the problem.

Dealing with real-life train driver scheduling implies that the resulting shifts must adhere to a large and complex set of labour agreement rules. This makes the task of assessing the feasibility of a potential shift generated by a move a hard one. To overcome this difficulty, local search interfaces in a black-box fashion with existing checking routines that are used in the generation phase of the GaS solver. This guarantees that all rules are enforced consistently throughout the algorithm.

The local search phase is not constrained by the set of shifts generated by GaS (as the selection phase in GaS is). We exploit this advantage in two ways. First, moves can work at a *piece-level* (i.e. the atomic unit on which spells and shifts are built upon). Second, it is usually the case that GaS is provided with certain parameters to restrict the size of the resulting pool of shifts; a typical example is restricting the length of spells. These parameters need not be enforced during the local search, allowing it to use shifts which would be deemed illegal during the generation phase of GaS. This adds another way of achieving improvements in the schedule, which is independent of the introduction of windows to the model.

Looping back to GaS

The schedule produced during the local search phase can be used to specify a new set of approximations for the windows, which can then be looped back to the GaS solver; moreover, any legal shifts generated during the local search can also be passed on to GaS. This provides the basis for an iterative procedure, in which each call to GaS results in a new initial solution for the local search, and each call to the local search yields a new set of approximations (and eventually an additional set of shifts) for a next call to GaS. Under this view, the local search acts as a tool for providing better windows' approximations to the GaS solver. This extension has proven to be very fruitful.

Experiments

The new algorithms are tested using real life data from a large rail network in Scotland. Preliminary versions of these algorithms have already produced operational schedules for this network, which improve on cost and/or reliability. There is ongoing work in obtaining new moves which tackle different aspects of schedule efficiency and reliability, refining the local search driving framework, and better exploiting the information gathered during the GaS and local search loop-back; results and conclusions will be discussed.

References

1. Kwan RSK (2004) Bus and train driver scheduling. In: Leung JYT (ed) Handbook of Scheduling: Algorithms, Models, and Performance Analysis, Chapter 51, CRC Press, FL, USA
2. Kwan ASK, Kwan RSK, Parker ME, Wren A (1999) Producing train driver schedules under different operating strategies. In: Wilson NHM (ed) Proceedings of the Seventh International Workshop on Computer-Aided Scheduling of Public Transport. 129–154, Springer-Verlag
3. Wren A, Fores S, Kwan ASK, Kwan RSK, Parker ME, Proll L (2003) A flexible system for scheduling drivers. Journal of Scheduling, vol. 6, 437–455
4. Shen Y, Kwan RSK (2001) Tabu search for driver scheduling Voß S, Daduna JR (eds) Computer-Aided Scheduling of Public Transport, 121–135, Springer-Verlag
5. Shen Y (2001) Tabu search for bus and train driver scheduling with time windows. PhD thesis, University of Leeds