Finding Delay-Tolerant Train Routings through Stations

Gabrio Caimi, Dan Burkolter, and Thomas Herrmann

Institute For Operations Research, Swiss Federal Institute of Technology,
Clausiusstrasse 47, 8092 Zurich, Switzerland
{caimig, danb, herrmann}@ifor.math.ethz.ch

Abstract. Currently, many railway operators are increasing the frequencies of their trains. By condensing the timetable, routing trains becomes increasingly difficult as the chosen routes not only have to meet safety restrictions, but also guarantee some stability if delays occur.

We address the problem of routing trains through railway stations for a given timetable and outline two algorithms. The first algorithm searches for a feasible solution for the train routing problem based on an independent set modeling that is solved using a fixed-point iteration method. The initial solution is then amended by applying the second algorithm in order to increase the time slot of a chosen route, i.e. the time interval during which a train may arrive and find its designated route open. This algorithm is based on a local search optimization scheme.

Results showed that the fixed-point iteration found feasible solutions within minutes even for difficult cases, i.e. tight timetables. Though more time-consuming, the second algorithm allowed the average time slot length to be doubled, thus implying that it is possible to find routings which are more delay-tolerant. This helps to decrease impacts of late trains.

1 Introduction

We consider the problem of routing trains through stations. As many railway operators are currently decreasing the cycle-time of their periodic timetables, finding a feasible and good train routing will become increasingly difficult. In particular, the problem of scarce capacity will be accentuated in major stations where it is desirable to provide for connections. As a consequence, trains tend to arrive and leave during a short interval.

We assume for the problem that a timetable and a layout of the station are given. The goal then is to find for each train a route from an entry point to a leaving point of the station area via its platform. Zwaneveld et al. have studied this problem in [1], [2], and [3]. They show that the problem is NP-complete and propose an independent set modeling that they solve with a Branch and Cut algorithm. We apply the same modeling, yet propose a fixed-point iteration heuristic for solving the problem as some of our test-instances are too large for a Branch and Cut algorithm.

To our knowledge, other researchers have so far only addressed the question of finding feasible routings. However, by choosing a good routing, the

impact of late trains on the timetable can be decreased. Ideally, the planned route for a train stays clear even if it has some delay and, as a consequence, operators need not intervene actively into the system. We introduce the concept of the time slot of a train route which shows how much time prior or later than the actual schedule a train may arrive and still use its planned route without coming into conflict with other running trains; see [4] for further information. We propose a local search optimization algorithm that increases the length of the time slots of routings starting from an initial feasible solution.

The paper is organized as follows. In section 2 an algorithm for finding an initial feasible solution and an algorithm for increasing the time slots are described. In section 3 computational results for a major station of the Swiss Federal Railways are given. Finally conclusions are drawn.

2 Model and Algorithm

We are given a set T of n trains each having a set $R_i = \{r_{i_1}, \dots, r_{i_{m(i)}}\}$, $i = 1, \dots, n$ of $m(i)$ possible routings connecting its entry and leaving point into the station area. Additionally, we have the conflict set C of incompatible routes, i.e. pairs of routes of different trains that at some place in the station topology use the same track segment at the same time (including safety time). We use the following notation for conflicting routes: $r_{p_q} \not\asymp r_{u_v}$. Analogously, $r_{p_q} \asymp r_{u_v}$ implies that the routes are compatible. Additionally, all routes of the same train are "in conflict" since only one route for each train is needed. Therefore, the conflict set is described by $C = \{\{r_{i_k}, r_{j_l}\} | i = j \vee r_{i_k} \not\asymp r_{j_l}\}$.

The routing problem can be seen as an independent set problem. The vertices are the elements of $R_1, \dots, R_n$ and the edges correspond to the elements of C. Note that within the resulting graph all routes of the same train build a clique. A feasible solution to the problem is now given by a maximum independent set. A tight upper bound on the maximum cardinality is known due to the clique structure of the graph. The maximum cardinality is equal to the number of trains if and only if it is possible to find a conflict-free routing for all trains.

2.1 Finding an Initial Solution

In order to find a maximum independent set, we apply a specialized version of a heuristic for solving *Constrained Semi-Assignment Problems* [5]. The basic idea of the heuristic is to make a continuous relaxation of the boolean decision variables and then evolve starting from an interior point towards an extremal point corresponding to a feasible assignment. Burkard et al. present theoretical results for the general algorithm and show by empirical comparisons with Tabu Search that the algorithm finds solutions to the *Constrained*