Discrete Applied Mathematics 161 (2013) 1707–1718

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

A Lagrangian heuristic for a train-unit assignment problem

Valentina Cacchiani, Alberto Caprara, Paolo Toth *

DEIS, University of Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy

ARTICLE INFO

Article history:
Received 18 October 2010
Received in revised form 20 October 2011
Accepted 30 October 2011
Available online 25 November 2011

Keywords:
Train-unit assignment
Heuristic
Lagrangian relaxation
Assignment problem
Local search

ABSTRACT

We present a fast heuristic for an important NP-hard problem, arising in the planning of a railway passenger system, that calls for the definition of the train units to be assigned to a given set of timetabled trips, each with a given number of passenger seats requested. The heuristic is based on the Lagrangian relaxation of a natural formulation of the problem, whose solution can be found by solving a sequence of assignment problems. With respect to an already existing method, the heuristic we propose turns out to be much faster in practice and still providing solutions of good quality. This makes it suitable for all cases in which the problem either must be solved many times, e.g., when it is integrated with other phases of railway planning, or when it must be solved within short computing time, e.g., within real-time operations.

1. Introduction

The optimization of a railway system is generally performed in separate phases; see e.g. [7]. In this paper, we focus on the phase called *Rolling Stock Planning*. Roughly speaking, it consists of finding the optimal assignment of *train units*, which are self-contained trains with an engine and passenger seats, to a given set of *trips* to be performed every day of the considered planning horizon and whose timetable has been specified in the previous *Timetabling* phase.

According to the requirements of the different railway companies, different versions of Rolling Stock Planning arise in the real-world. For instance, in our case study, we have to decide for each train-unit the sequence of the corresponding assigned trips, while in the case of NS (the main Dutch Train Operator company) for each trip the successor trip is given on input. On the other hand, when combining more than one train unit for covering a trip, here we do not consider the order of the train units, while in the NS case this must be taken into account [10,14]. A detailed literature review is out of the scope of this paper as it can be already found in [6,7]. In any case, there are many papers dealing with the assignment of train units to trips [1,2,4,10,16,17]. In other cases the problem consists in finding an optimal assignment of locomotives and cars, instead of self-contained train units, to the given set of trips [5,8,9,12]. In all these references, different heuristic and exact approaches are described, mostly based on *Integer Linear Programming* (ILP) formulations solved by either general-purpose solvers, branch-and-price, Benders' decomposition, or branch-and-cut. In some cases uncertainty is taken into account by using stochastic optimization techniques.

In the recent years, approaches that *integrate* two consecutive phases of the railway system optimization were developed; see e.g. [3,13,15]. For instance, consider the potential integration of the Timetabling phase with the Rolling Stock Planning phase: the various candidate timetables produced in the former may be evaluated also by taking into account the quality of the solution obtained for the latter. To this aim, it is essential to have fast and effective algorithms for Rolling Stock Planning, to be applied for every candidate Timetabling solution.

In addition, the study of *real-time* solutions has become more and more important in railway optimization. In particular, if a disruption takes place, a solution that has been obtained in the planning phase might become infeasible at an operational

* Corresponding author. Fax: +39 051 2093073.
E-mail addresses: valentina.cacchiani@unibo.it (V. Cacchiani), alberto.caprara@unibo.it (A. Caprara), paolo.toth@unibo.it (P. Toth).

0166-218X/$ – see front matter
doi:10.1016/j.dam.2011.10.035

level. In this case, a *recovery* algorithm has to be executed to recompute a good feasible solution with the new given data. Also in this situation, fast and effective algorithms are needed.

The contribution of this paper goes in this direction: we develop a fast and effective heuristic algorithm for our case study. We present computational experiments comparing, on real-world instances of a regional Italian Train Operator, the new algorithm with the previous much more sophisticated approach proposed in [6]. The computational experiments clearly show that the new algorithm has good behavior both in terms of computing time and quality of the solution.

It is interesting to note that the new heuristic is based on the natural Lagrangian relaxation of a natural Integer Linear Programming (ILP) formulation of the problem, which we already considered in [6] where we summarize the outcome as follows "…our implementation of a customary heuristic method based on this Lagrangian relaxation performed extremely poorly in practice for our case study, in terms of both lower bound produced and solution found…Given that the results were so poor, we do not even present these results". What we missed in [6] was an effective definition of the score to use in the construction of a solution and the use of local search procedure to improve the solution constructed.

The paper is organized as follows. In Section 2, we formally describe the problem in our case study and present its canonical representation on a graph. In Section 3, we propose an ILP formulation with arc variables, a heuristic algorithm, based on the Lagrangian relaxation of two sets of constraints and on the decomposition relaxation of another set of constraints, and a local search procedure. In Section 4, we present computational experiments on a set of real-world instances. Conclusions and guidelines for future research are discussed in Section 5.

2. The Train-Unit Assignment problem

We will call the specific Rolling Stock Planning problem in our case study *Train-Unit Assignment* (TUA). Its input consists of a set of train trips and a set of train-unit types. Each trip has a departure station, an arrival station, a departure time, an arrival time, and a *seat request*, given by the estimated number of passengers traveling on the trip. Each train-unit type has a number of available units and a *capacity*, given by the number of available seats for each unit. Train units can be combined with each other in order to fulfill the seat requests of the trips. We say that a trip is *covered* when its seat request is fulfilled and *uncovered* otherwise.

We have to decide which of the available train units to use, and to assign each train unit used a sequence of trips to be performed within the same day by the unit. In reality, each physical train unit of a given type will perform on each day of the time horizon the daily sequence of trips assigned to some train unit of that type in the TUA solution. This generally results in daily sequences of trips for physical train units that are repeated cyclically over the time horizon. Given that defining such cyclic sequences is easy after having defined the daily sequences in TUA (see also [6]), it is natural to focus only on the latter. The objective is to minimize the number of train units used subject to the following constraints:

- *covering*: the seat request must be satisfied for each trip;
- *combination*: a maximum number of train units can be combined in order to cover each trip;
- *sequencing*: two trips can be performed in sequence by a given train unit in the same day if and only if there is enough time for the train unit for traveling from the arrival station of the first trip to the departure station of the second trip (note that the traveling time between two stations generally depends on the type of the considered unit);
- *availability*: no more units than the ones available for each type can be used.

Formally, let n be the number of trips and p the number of train-unit types. For each trip $j = 1, \dots, n$, let r_j be the seat request, t_j be the *departure time* and u_j be the maximum number of train units that can be assigned to the trip. For each train-unit type $k = 1, \dots, p$, let d^k be the number of available train units and s^k the capacity. As discussed in [6], if needed we redefine (increase) the seat request r_j of each trip j so that it is always possible to find a combination of train units whose overall capacity is exactly r_j. This preprocessing procedure applied to the seat requests is very simple and fast, leads to stronger lower bounds, and is important for both the constructive and the local search phase of the algorithm we propose. Indeed, it is useful to know how "well" a train unit matches the seat request of a trip in order to determine which are the best trips to be assigned to the train unit. Thus, if a trip j has a seat request r_j that is not equal to any combination of up to u_j train unit capacities s^k, then we redefine its seat request, considering that the number of seats assigned to it in the TUA solution will always be higher than r_j.

We introduce a directed complete multigraph $G = (V, A)$ to represent the problem. The node set V corresponds to the set of trips, and the arc set A is partitioned into p subsets $A^1, \dots, A^p$, where arc subset A^k is associated with train units of type k and the simple directed graph $G^k = (V, A^k)$ is complete. The sequencing constraints are implicitly represented by the costs of the arcs. In particular, if trip j can be performed right after trip i by a train unit of type k in the same day, the cost c_{ij}^k of arc $(i, j)^k \in A^k$ is given by the time in minutes elapsing between the departure of trip i and the departure of trip j (i.e., $c_{ij}^k = t_j - t_i$). Otherwise, the cost c_{ij}^k of arc $(i, j)^k \in A^k$ is given by the time in minutes elapsing between the departure of trip i and the departure of trip j *on the next day* (i.e., $c_{ij}^k = t_j - t_i + 1440$). In this way, each cycle in G^k has a cost that is an integer multiple of 1440 (the number of minutes in a day), say $1440q$, and corresponds to the trips that can be assigned to q train units of type k. Therefore, the problem calls for a minimum-cost collection of cycles in G such that the total length of the cycles in A^k does not exceed $1440d^k$ and each vertex j is visited by at most u_j cycles with overall capacity at least r_j (the