A Heuristic Algorithm for the Train-Unit Assignment Problem

V. Cacchiani * A. Caprara P. Toth

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

Key words: Train Unit Assignment, Lagrangian relaxation, Heuristic, Assignment Problem

1 The Train-Unit Assignment Problem

We study the Train-Unit Assignment Problem (TUAP) (see e.g. [2]), calling for an optimal assignment of train units (which are self-contained trains with an engine and passenger seats) to a given set of timetabled train trips. More precisely, each trip has a departure station, an arrival station, a departure time and an arrival time, and requires a number of passenger seats. Train units can be classified in different types: each train unit has a number of available seats and can be combined with other units in order to fulfill the seat requests. For each train trip a maximum number of train units that can be combined is given. In addition, sequencing constraints between trips must be satisfied: a pair of trips can be in a sequence for a train unit if the time elapsing between the arrival of the first one and the departure of the second one is large enough to allow the corresponding train unit to travel from the arrival station of the first one to the departure station of the second one. Finally, each train unit has to undergo a maintenance operation every fixed number of days, which requires a certain amount of time, as well as the transfer to and from the maintenance station. The goal is to minimize the number of train units globally used, while satisfying the seat requests, the sequencing constraints and the maintenance constraints.

* V. Cacchiani
Email addresses: valentina.cacchiani@unibo.it (V. Cacchiani), alberto.caprara@unibo.it (A. Caprara), paolo.toth@unibo.it (P. Toth).

2 Solution method

Let n be the number of trips and p the number of train unit types. Each trip $j \in \{1, \ldots, n\}$ is defined by a request r_j, given by the required number of passenger seats, a maximum number u_j of train units that can be assigned to the trip, and its timetable. Each train unit type $k \in \{1, \ldots, p\}$ is defined by a number d^k of available train units and an associated capacity s^k, given by the number of available seats. We consider the natural and well-known Integer Linear Programming (ILP) with arc variables, based on a canonical graph representation of the problem (see e.g. [1]). Let $G = (V, A)$ be a directed acyclic multigraph, in which nodes correspond to trips, and the arc set A is partitioned into p subsets $A^1, \ldots, A^p$, where A^k is associated with train units of type k. The sequencing constraints are implicitly represented by the graph. In particular, arc $(i, j)^k$ exists whenever the time between the arrival of trip i and the departure of trip j allows a train unit of type k to travel from the arrival station of trip i to the departure station of trip j. Let us introduce an integer variable x^k_{ij} ($k \in \{1, \ldots, p\}$, $i, j \in \{1, \ldots, n\}$), that indicates the number of times that arc $(i, j)^k$ is selected in the solution, i.e., the number of train units of type k that execute trip i before trip j in the associated sequence. Moreover, let c^k_{ij} denote the cost of arc $(i, j)^k$, corresponding to the time in minutes elapsing between the departure of the starting node and the departure of the ending node. The ILP reads as follows:

$$\min \sum_{k=1}^{p} \sum_{i=1}^{n} \sum_{j=1}^{n} c^k_{ij} x^k_{ij}, \tag{1}$$

$$\sum_{i=1}^{n} x^k_{ij} = \sum_{i=1}^{n} x^k_{ji}, \quad k \in \{1, \ldots, p\},\ j \in \{1, \ldots, n\}, \tag{2}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c^k_{ij} x^k_{ij} \leq 1440\ d^k, \quad k \in \{1, \ldots, p\}, \tag{3}$$

$$\sum_{k=1}^{p} \sum_{i=1}^{n} s^k x^k_{ij} \geq r_j, \quad j \in \{1, \ldots, n\}, \tag{4}$$

$$\sum_{k=1}^{p} \sum_{i=1}^{n} x^k_{ij} \leq u_j, \quad j \in \{1, \ldots, n\}, \tag{5}$$

$$x^k_{ij} \geq 0,\ \text{integer}\ , \quad k \in \{1, \ldots, p\}, i, j \in \{1, \ldots, n\}. \tag{6}$$

The objective is to minimize the overall cost of the selected arcs, equal to 1440 (the number of minutes in a day) times the number of the train units globally used. Constraints (2) express the flow conservation. Constraints (3) forbid to use more than the available train units for each type. Constraints (4) impose to cover each trip with the corresponding seat request. Constraints (5) impose a bound on the number of train units that can be used to cover each trip.

Note that the maintenance constraints are not introduced in the formulation, since they would be complex to model; we take them into account directly in the heuristic algorithm.

The approach in [1], based on solving LP relaxations by general-purpose LP software, considers an alternative equivalent ILP formulation with path variables, whose LP relaxation is much faster to solve (modulo using column generation). In this work, we consider a Lagrangian-relaxation based approach, which nicely combines with the ILP above. Specifically, we first replace constraints (5) by the weaker

$$\sum_{i=1}^{n} x_{ij}^{k} \leq u_j, \quad k \in \{1, \ldots, p\}, j \in \{1, \ldots, n\}, \tag{7}$$

which impose that each train unit of type k can cover each trip j at most u_j times. Then, we relax in a Lagrangian way constraints (3) and (4), by using nonnegative Lagrangian multipliers $\lambda_j, (j \in \{1, \ldots, n\})$ and $\sigma_k, (k \in \{1, \ldots, p\})$, respectively.

The resulting Lagrangian relaxed problem decomposes onto p independent subproblems, one for each train unit type. Let $\tilde{c}_{ij}^{k} := (c_{ij}^{k} - \lambda_j s^k + \sigma_k c_{ij}^{k})$ be the Lagrangian cost for each arc $(i,j)^k \in A$. We impose a null Lagrangian cost ($\tilde{c}_{ii}^{k} := 0$) for loops $(i,i)^k$ for each vertex $i \in V$ (that had originally infinity cost). Thus, the subproblem associated with a train unit type k calls for the minimization of $\sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{c}_{ij}^{k} x_{ij}^{k}$ subject to (2) and (7). Moreover, the presence of zero-cost loops allows us to replace inequality by equality in (7). It is well known that all vertices of the feasible region of this subproblem are integer for u_j integer and the associated constraint matrix is totally unimodular. Moreover, in the particular case, arising in our case study, in which u_j does not depend on j, we can replace u_j with u for $k \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, n\}$. This makes the subproblem equivalent to an Assignment Problem (AP), obtained by replacing u by 1 in (7), the correspondence between solutions $\bar{x}$ of the subproblem and $\bar{y}$ of AP being given by $\bar{x} = u\bar{y}$. As is well known, the assignment problem can be solved in $O(n^3)$ time. In order to find good Lagrangian multipliers we apply a standard iterative subgradient procedure.

Besides yielding a valid lower bound on the optimal solution value, the best Lagrangian multipliers λ^*, σ^* found throughout the iterations and the corresponding reduced costs $\bar{c}_{ij}^{k} := c_{ij}^{k} - \lambda_j^* s^k + \sigma_k^* c_{ij}^{k} - \bar{w}_i^k - \bar{v}_j^k$, where $\bar{w}^k$ and $\bar{v}^k$ are the optimal AP dual variables associated with the assignment constraints, are used to drive the following constructive Lagrangian heuristic algorithm. We order the train unit types for decreasing capacity values, and construct the workload for each train unit, until either all the trips have been covered or all the train units have been used (in this case, we apply a local search procedure to obtain a feasible solution). The construction starts with the selection of a trip whose departure is in the beginning of the day. Then, we choose the following trips by assigning to each one a score that takes into account the reduced costs, the original costs and how “well” the current train unit can cover the trip (taking into account which other train units are still available). In addition, we give a prize to the arcs that allow to perform maintenance, until we reach the number of necessary maintenance operations. Every time a trip is selected in the solution, we update its number of seats and the reduced cost of each arc entering the corresponding node. In order to decide how to end the workload of the current train unit type, for each selected trip in the workload we solve the relaxed problem on the residual train units and trips. The solution value of the reduced relaxed problem gives a lower bound on the global number of train units that are needed to cover all the residual trips. When no more train units are available of the current type, we end the workload with the trip giving the smallest solution value.

3 Computational Experiments

We present some preliminary computational experiments on a set of real-world instances for an operator running trains in a regional area, and compare the results with the approach presented in [1], with and without imposing the maintenance constraints. The tests were performed on a PC Pentium 4, 3.2 GHz, 2 GB RAM, and using Cplex 9.0 as an LP-solver in [1]. The results are presented in Table 1, showing that the proposed approach produces comparable results within much shorter computing time (expressed in seconds).

			Lagr. heur.		[1] heur.		Lagr. heur. maint.		[1] heur. maint.	
inst.	n	p	value	time	value	time	value	time	value	time
1	85	1	2	0	2	0	2	0	2	0
2	120	1	4	0	4	0	4	1	4	5
3	302	1	17	4	17	288	17	5	17	544
4	208	2	26	4	25	17	27	3	25	19
5	364	2	20	18	20	1912	21	20	20	3899

Table 1
Comparison on a set of real-world instances in the case without or with the maintenance constraints.

References

[1] Cacchiani, V., Caprara, A., and Toth, P., “Solving a Real-World Train Unit Assignment Problem”, *Mathematical Programming Series B*, to appear (2010).

[2] Caprara, A., Kroon, L., Monaci, M., Peeters, M., and Toth, P., “Passenger Railway Optimization”, in C. Barnhart and G. Laporte (eds.), *Handbooks in OR & MS*, Vol. 12, Elsevier Science (2006).