A Mixed Integer Linear Program for Optimizing the Utilization of Locomotives with Maintenance Constraints

Sarah Frisch* Philipp Hungerländer† Anna Jellen‡ Dominic Weinberger§

September 10, 2018

Abstract

In this paper we investigate the Locomotive Scheduling Problem, i.e., the optimization of locomotive utilization with prior known transports that must be performed. Since railway timetables are typically planned a year in advance, the aim is to assign locomotives to trains such that the locomotive utilization is maximized while maintenance constraints are taken into account. We model this optimization problem on a sparse weighted directed graph that defines the input variables for our proposed Mixed Integer Linear Program (MILP). We consider two different objective functions: First we minimize over the number of deadhead kilometers, i.e., kilometers from a locomotive driven without pulling a train, and second, over the number of locomotives used. In a further step, we combine the two suggested objective functions, i.e., we minimize over the weighted sum of them. Finally, we conduct a computational study to compare the performance of our MILP with the different proposed objective functions and show how the MILP can be used within a rolling horizon approach.

Keywords: Locomotive scheduling problem, maintenance constraints, mixed integer linear programming.

1 Introduction

This paper discusses a variant of the Locomotive Scheduling Problem (LSP), where the aim is to assign a fleet of locomotives to a set of scheduled trains, such that the overall costs are minimized. The rolling stock represents one of the main costs of a rail transportation system. The costs are composed of the one-time investment for the acquisition and running costs for fuel and maintenances. Our aim is to reduce these costs by minimizing either the number of locomotives used, the number of deadhead kilometers, or the combination of both. After giving an overview of already existing approaches in locomotive scheduling in Section 2, we state our specific problem and model it on a sparse weighted directed graph in Section 3. We continue with formulating a Mixed Integer

*Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Austria, sarah.frisch@aau.at

†Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Austria, philipp.hungerlaender@aau.at

‡Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Austria, anna.jellen@aau.at

§Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Austria, dominic.weinberger@hex-solutions.com

Linear Program (MILP) based on the problem graph in Section 4. Finally, we conduct a computational study on a part of the Austrian railway network in Section 5. The aim of our proposed optimization model is to test the different objective functions relating to runtime and quality of the solution. We further test our MILP within a rolling horizon approach. We show that our MILP delivers high-quality solutions for the LSP in freight transportation and is applicable by the Rail Cargo Austria (RCA).

2 Literature Review

Optimization in rail transportation is a widely studied field. Vaidyanathan et al. [5] formulate the LSP as an integer-programming string decomposition problem on a suitably defined space-time network considering fueling and maintenances. They solve it by an aggregation-disaggregation based algorithm. Noori et al. [4] consider maintenances but in contrast to our approach the trains in their model have different priorities to be conducted. They model a Vehicle Routing Problem with Time Windows, where the trains act as costumers, and solve it heuristically. Reuther et al. suggest several algorithms to solve the rolling stock rotation problem, i.e., the LSP, where a cyclic planning horizon over one week due to a periodic input schedule is considered. They state a hypergraph based Integer Program (IP) formulation in [1], which is extended for considering maintenances in [2]. They solve the IP by using an algorithm based on column generation and a method called rapid branching. In [3] they provide a local search heuristic to solve large scale instances for the DB Fernverkehr AG and ICE vehicles. In contrast to [1] where hyperarcs are introduced, we define a sparse simple graph and based on this graph we propose a MILP that we solve without applying heuristics. We minimize over deadhead kilometers and locomotives used, and provide three objective functions. This differs from [5] where costs for maintenances and ownership are minimized, and from [1] where costs for locomotives, services, deadhead kilometers, regularity, and coupling are minimized. Compared to the approaches mentioned above our MILP is applied on precise calender dates and within a rolling horizon approach.

3 Problem Description and Modeling

In this section we formulate our problem by defining a graph that will later build the basis for our MILP. We assume to have m locomotives $F = \{f_1, \dots, f_m\}$ at disposal. Each $k \in F$ is allowed to travel $r_k^{max} \in \mathbb{R}^+$ kilometers until a maintenance must be conducted. We have a schedule of trains that must be performed with information about stations and time for departure and arrival. We introduce a directed, weighted graph $D = (V, A_1, A_2, A_3)$, where V is the set of nodes and A_1, A_2, and A_3 are sets of edges. Nodes represent stations at a particular time, i.e., Station 1 at 10:00 am and Station 1 at 12:00 pm are represented by two different nodes. Edges display all kinds of trips that a locomotive is able to perform, A_1 represents scheduled trips, A_2 represents deadhead trips, and A_3 represents deadhead trips including a maintenance. We define two nodes for each scheduled trip: a *departure node* $v \in V^d \subset V$ and an *arrival node* $v \in V^a \subset V$.

For each scheduled trip we draw an edge adjacent to the corresponding departure and arrival nodes. Furthermore, we connect edges of scheduled trips by edges that represent deadhead trips with or without maintenances. The allocation of locomotives can be illustrated as follows: Locomotives travel alternating on scheduled trips and deadhead trips with or without maintenances, such that each node (station at a specific time) is visited exactly once by exactly one locomotive.

In order to meet all constraints, we enlarge the set of nodes and refine the definitions of the graph. At the beginning of the planning horizon each locomotive has mileage r_k^s and is placed at its *start node* $v_k \in V^s \subset V$, a station at the time of the beginning of the planning horizon. We also introduce an artificial *final node* v_f and define the set $\{v_f\} =: V^f \subset V$. We draw an edge for a deadhead trip with distance zero, from each arrival node to the final node. In this way we are able to model (in our MILP) that all locomotives can end in an arbitrary station. In summary the set of nodes is $V = V^d \cup V^a \cup V^s \cup V^f$. Now let us also give a more detailed description of the sets of edges:

a) *The set of edges for scheduled trips* A_1*:* For each scheduled trip we define a directed edge $(i, j) \in A_1 \subseteq V \times V$. The edge is weighted by the distance d_{ij} between the corresponding departure and arrival nodes. In order to provide enough power for pulling the trains, only some locomotives are allowed to conduct certain trains. Hence we introduce the set $P_{ij} = \{k \in F : k \text{ is permitted to conduct the trip from } i \text{ to } j\}$.

b) *The set of edges for deadhead trips* A_2*:* For a possible deadhead trip we introduce a directed edge $(i, j) \in A_2 \subseteq V \times V$ that is weighted by the distance d_{ij} between the corresponding stations of Nodes i and j. We call Node j *reachable* for a locomotive starting in Node i, if there exists a path between the corresponding stations and if the time at i plus the travel duration is smaller than the time at j. We draw such edges between an arrival and a reachable departure node, a start node and each reachable departure node, and each start, respectively arrival, node and the final node.

c) *The set of edges for deadhead trips with maintenances* A_3*:* For a deadhead trip with maintenance we define a directed edge $(i, j) \subset V \times V$. As for edges in A_1 we have a set P_{ij} that contains all locomotives that are permitted to conduct the trip from i to j. For edges of deadhead trips with maintenances P_{ij} contains exactly one element $k \in F$. Therefore we obtain edges of the form $(i, j, k) \in A_3 \subseteq V \times V \times F$, that are weighted by the distance d_{ij} between the corresponding stations of Nodes i and j. We draw such an edge from i to j for a $k \in F$, if all conditions for an edge in A_2 are satisfied, and if additionally the time at i plus the travel duration plus the time for the maintenance of k is smaller than the time at j, and k is admitted to be maintained in j.

Finally let us introduce a subset of nodes that is needed for the MILP formulation: $V_k^{s+1} \subset V$ contains all nodes $v \in V \setminus V^f$ such that there is an edge in A_2 or A_3 connecting the start station v_k of $k \in F$ and v. In total we obtain a sparse graph with the advantage that for the corresponding MILP it is not necessary to introduce variables for explicitly modeling the time component of the trains.

4 Mathematical Formulation

Based on the described Graph D we formulate a Mixed Integer Linear Program (MILP) with different objective functions. We start with minimizing the overall deadhead kilometers and introduce the following decision variables:

- $x_{ij} \in \{0, 1\}$, $(i, j) \in A_2$, with $x_{ij} = 1$, if edge (i, j) is used.
- $y_{ijk} \in \{0, 1\}$, $(i, j, k) \in A_3$, with $y_{ijk} = 1$, if edge (i, j, k) is used.
- $q_{ik} \in \{0, 1\}$, $i \in V \setminus V^f$, $k \in F$, with $q_{ik} = 1$, if Node i is visited by Locomotive k.
- $r_i \in \mathbb{R}^+$, $i \in V$, denotes the mileage in Node i.

$$\min \quad \sum_{(i,j)\in A_2} d_{ij}x_{ij} + \sum_{(i,j,k)\in A_3} d_{ijk}y_{ijk} \tag{Objective I}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A_2} x_{ij} + \sum_{(i,j,k)\in A_3} y_{ijk} = 1, \quad j \in V^d, \tag{1}$$

$$\sum_{(i,j)\in A_2} x_{ij} + \sum_{(i,j,k)\in A_3} y_{ijk} = 1, \quad i \in V^a \cup V^s, \tag{2}$$

$$\sum_{(i,j)\in A_2} x_{ij} = m, \quad j \in V^f, \tag{3}$$

$$q_{ik} = 1, \quad i \in V^s,\ k \in F,\ i = k, \tag{4}$$

$$\sum_{k\in F} q_{ik} = 1, \quad i \in V \setminus V^f, \tag{5}$$

$$\left.\begin{array}{l} q_{ik} = q_{jk}, \quad (i,j) \in A_1,\, k \in F,\ \text{if } k \in P_{ij}, \\ q_{ik} + q_{jk} = 0, \quad (i,j) \in A_1,\, k \in F,\ \text{if } k \notin P_{ij}, \end{array}\right\} \tag{6}$$

$$q_{jk} \geq q_{ik} - (1 - x_{ij}), \quad (i,j) \in A_2,\, k \in F,\, j \notin V^f, \tag{7}$$

$$y_{ijk} \leq q_{\ell k}, \quad (i,j,k) \in A_3,\, \ell \in \{i,j\}, \tag{8}$$

$$r_i = r_i^s, \quad i \in V^s, \tag{9}$$

$$r_j = r_i + d_{ij}, \quad (i,j) \in A_1, \tag{10}$$

$$r_j \geq r_i + d_{ij} - M_1(1 - x_{ij}), \quad (i,j) \in A_2, \tag{11}$$

$$r_j \leq M_2(1 - y_{ijk}), \quad (i,j,k) \in A_3, \tag{12}$$

$$r_i \leq r_k^{max} + M_3(1 - q_{ik}), \quad i \in V \setminus V^f,\ k \in F, \tag{13}$$

$$r_i + d_{ijk} \leq r_k^{max} + M_3(1 - y_{ijk}), \quad (i,j,k) \in A_3, \tag{14}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in A_2, \tag{15}$$

$$y_{ijk} \in \{0,1\}, \quad (i,j,k) \in A_3, \tag{16}$$

$$q_{ik} \in \{0,1\}, \quad i \in V \setminus V^f,\ k \in F, \tag{17}$$

$$r_i \geq 0, \quad i \in V. \tag{18}$$

Constraints (1) and (2) guarantee that each departure node has exactly one outgoing edge and each arrival respectively start node has exactly one ingoing edge. Equations (3) secure that all locomotives enter the final node. Constraints (4) link locomotives with their departure stations. Equations (5) ensure that except for the final station every station is visited by exactly one locomotive. Constraints (6) guarantee that only permitted locomotives conduct scheduled trips. Inequalities (7) ensure for $(i,j) \in A_2$ that Nodes i and j are visited by the same locomotive. Constraints (8) secure that $(i,j,k) \in A_3$ can be used for a locomotive just if it has visited the corresponding departure and arrival node. The mileages are initialized and updated in Constraints (9) and (10). The mileage of deadhead kilometers is updated by Inequalities (11). The mileages after performed maintenances are initialized by Constraints (12). Inequalities (13) and (14) avoid to exceed the maximal permitted mileage of a locomotive until the next maintenance. We recommend to choose $M_1 = \max_{k\in F} r_k^{max} + \max_{(i,j)\in A_2} d_{ij}$, $M_2 = \max_{k\in F} r_k^{max}$, and $M_3 = \max_{(i,j,k)\in A_3} d_{ijk}$.

For minimizing over the number of locomotives we introduce additional decision variables $s_k \in \{0,1\}$, $k \in F$, with $s_k = 1$, if Locomotive k is used at least once within the considered planning horizon. We alter the MILP by replacing (Objective I) by (Objective II) and adding Constraints

(19).

$$\min \quad \sum_{k \in F} s_k \qquad \text{(Objective II)}$$

$$\text{s.t.} \quad s_k \geq \left(\sum_{j \in V_k^{s+1}} x_{ij} + y_{ijk} \right), \quad i \in V^s,\ k \in F. \qquad (19)$$

Finally we suggest to combine the two proposed objective functions. To do so we run our two MILPs and use their objective values, denoted by z_{km} and z_{loc}, as weights to gain a good trade-off. We alter our second MILP by replacing (Objective II) by (Objective III):

$$\min \quad z_{loc}\left(\sum_{(i,j) \in A_2} d_{ij} x_{ij} + \sum_{(i,j,k) \in A_3} d_{ijk} y_{ijk} \right) + z_{km} \sum_{k \in F} s_k \qquad \text{(Objective III)}$$

5 Computational Study

To simulate a real world scenario regarding the RCA's daily operations we tested our three MILPs on a part of the Austrian railway network. We created train schedules regarding an approximated traffic density and used real-world parameters for the specific dates and durations of maintenances. Considering a planning horizon of a week we solved each problem entirely, and then applied our MILPs within a rolling horizon approach by splitting the time interval into half. Our experimental results are summarized in Table 1.

We observe that the more locomotives are at disposal, the less deadhead kilometers are driven. But the CPU time increases due to the sharply rising number of edges in A_3. Considering the whole time interval the corresponding MILPs deliver high quality solutions, but they are too time consuming for large instances. By splitting the interval we achieve excellent solution times and an acceptable solution quality. Results under (Objective III) show a notably good trade-off between the number of locomotives used and the number of deadhead kilometers.

As future work we intend to reduce Sets A_2 and A_3 and thus the variables of our MILP by permitting corresponding edges just within a certain time window. The proposed model could be expanded by capacities of maintenance and train sections. Minimizing the overall number of maintenances conducted could be considered as another objective.

		whole time interval			splitted time interval		
#locs/#trains	obj	CPU time	LU	DH	CPU time	LU	DH
40/80	I	04:03	34	47.77	00:08	31	52.57
	II	21:27	20	268.08	00:18	27	313.57
	III	(29:06) 03:36	24	51.37	(00:35) 00:09	27	61.62
40/120	I	18:18	39	68.91	00:25	36	75.27
	II	52:48	29	387.80	00:47	36	474.01
	III	(01:23:34) 12:28	32	70.53	(01:42) 00:30	36	80.34
60/60	I	02:11	38	36.16	00:11	37	40.06
	II	28:32	15	193.52	00:02	21	203.65
	III	(34:50) 04:07	19	40.75	(00:26) 00:13	26	45.90
60/80	I	12:07	45	46.15	00:14	41	50.17
	II	14:33	20	257.18	00:31	33	299.80
	III	(34:54) 08:14	24	49.75	(00:51) 00:06	32	56.53
60/120	I	39:57	50	46.57	00:57	50	60.32
	II	TO	-	-	03:54	41	452.65
	III	-	-	-	(05:56) 01:05	42	66.41

Table 1: We display the solution time (mm:ss), the number of locomotives used (LU), and the number of deadhead kilometers (DH) of the MILP under each proposed objective. The time within the braces displays the overall solution time of all three MILPs. The time out (TO) per MILP was set to an hour. All experiments were performed using Gurobi 7.5.2 on a Linux 64-bit machine equipped with Intel(R) Xeon(R) CPU e5-2630 v3@2.40GHz and 128 GB RAM in multi thread mode with four cores.

References

[1] R. Borndörfer, M. Reuther, T. Schlechte, and S. Weider. A hypergraph model for railway vehicle rotation planning. *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, 2011.

[2] R. Borndörfer, M. Reuther, T. Schlechte, and S. Weider. Vehicle rotation planning for intercity railways. *Proceedings of Conference on Advanced Systems for Public Transport*, 2012.

[3] R. Borndörfer, M. Reuther, T. Schlechte, K. Waas, and S. Weider. Integrated optimization of rolling stock rotations for intercity railways. *Transportation Science*, pages 50(3):863–877, 2016.

[4] S. Noori and S. Ghannadpour. Locomotive assignment problem with train precedence using genetic algorithm. *Journal of Industrial Engineering International*, 2012.

[5] B. Vaidyanathan, R. K. Ahuja, and J. B. Orlin. The locomotive routing problem. *Transportation Science*, pages 42(4):492–507, 2008.