Combinatorial Optimization Model for Railway Engine Assignment Problem

Tibor Illés[1], Márton Makai[2], and Zsuzsanna Vaik[3] *

[1] Eötvös Loránd University of Sciences, Department of Operations Research, H-1117, Budapest, Pázmány Péter sétány 1/C, Hungary
illes@math.elte.hu
[2] Eötvös Loránd University of Sciences, Department of Operations Research, H-1117, Budapest, Pázmány Péter sétány 1/C, Hungary
marci@cs.elte.hu
[3] Szent István University, Ybl Miklós Architecture College, Department of Mathematics
H-1146, Budapest, Thököly út 74, Hungary
zsuzska@cs.elte.hu

Abstract. This paper presents an experimental study for the Hungarian State Railway Company (MÁV). The engine assignment problem was solved at MÁV by their experts without using any explicit operations research tool. Furthermore, the operations research model was not known at the company. The goal of our project was to introduce and solve an operations research model for the engine assignment problem on real data sets. For the engine assignment problem we are using a combinatorial optimization model. At this stage of research the single type train that is pulled by a single type engine is modeled and solved for real data. There are two regions in Hungary where the methodology described in this paper can be used and MÁV started to use it regularly. There is a need to generalize the model for multiple type trains and multiple type engines.

Keywords. Engine assignment, circulation

1 Introduction

The area of railway operation involves a lot of deep optimization problems. The following real-life optimization problem was addressed by the Hungarian State Railway Co. Pl., (MÁV). The timetable of passenger trains of a region of Hungary is given, and engines (locomotives) should be assigned to each passenger train under some operational policies. The timetable contains all the necessary data related to the train like departure and arrival stations and times, railway lines where the trains are operated, etc. In the given region exactly one type of

* Research is supported by OTKA grants T 037547 and TS 049788, by European MCRTN Adonet, Contract Grant No. 504438 and by the Egerváry Research Group of the Hungarian Academy of Sciences.

ATMOS 2005
5th Workshop on Algorithmic Methods and Models for Optimization of Railways
http://drops.dagstuhl.de/opus/volltexte/2006/662

train is operated, with one type of locomotives. This information simplifies the problem. Our goal is to find an assignment of the engines to the trains that uses a minimum number of engines.

The operational policy of MÁV includes for instance that every engine between 48 to 60 hours of running must go for maintenance at one station of a prespecified subset of stations. Furthermore, it is allowed to define an artificial train with the goal to send a locomotive from a station to another station if this locomotive is needed to complete the assignment for a day (or a period) at that station. It can happen that by adding some artificial trains to the timetable we might find assignments that use less locomotives than the solution of the problem with the original timetable. Therefore from a practical point of view we might have different objectives like solving the problem with a minimum number of locomotives or minimizing the total energy consumption of the locomotives used in the assignment or minimizing the operational cost of the passenger trains over that region (with or without some artificial trains included). Currently, for the experts of the MÁV, solving this problem with the criteria of minimizing the number of necessary locomotives in the assignment, takes a few days for the given test region. Our task is to find a mathematical model for the problem and generate optimal solutions quickly.

After some attempts to collect the necessary data for our modeling purposes we agreed with the experts of MÁV that the goal of the optimization is to find the minimum number of necessary locomotives. However, we believe that if we had all necessary data we could have all the above mentioned objective functions for optimization.

Therefore the problem is the following: a passenger train timetable for a region of Hungary and a set of engines are given, and we have to assign an engine to each train so that each engine can be assigned to at most one train at a time. The connection of the engine from a train, which has just arrived at the station to the next train takes about half an hour. This time is known as *connection time* and its exact value influences the assignment of engines to the trains at a given station. The engines must go for maintenance at one station of a prespecified subset of the stations within a prescribed interval of working hours. After analyzing the problem it is clear that this maintenance condition makes the problem NP-complete. The task is again to minimize the number of used engines.

The MÁV has many different types of engines. For our purpose, we only have distinguished diesel-engines and electric-engines. As the lines of these two types are disjoint for technical reasons, we can assume that we have one type of engine, and each engine can pull each train. This assumption simplifies our model.

The problem refers to the classical engine assignment problem which is one of the most important problems of railway optimization (see e.g. Ahuja et al. [1]). Although this seems to be a widely studied problem, the instances with different extra conditions and specifications require different combinatorial optimization models and methods. These distinctions appear also in the size of the solvable instances. So there are locomotive-car assignment problems with different conditions that can be written as (mixed)-integer programs, see for example Cordeau et al. [2], [3]. Most of these are solved by general optimization techniques as LP-relaxation, Branch-and-Bound method or Bender's decomposition.

We built up a 0-1 programming model, presented in [4], which has a lot of variables and conditions but it seems to describe the complete problem. This integer programming model was too large to be used for computations, (and unfortunately integer programming models and LP solvers are not widely used at MÁV by their experts). So we consider the engine assignment problem without the maintenance condition, that is called the weak engine assignment problem ($WEAP$). This problem can be modeled as a purely combinatorial optimization one, that is to find a minimum number of paths in a graph, which are disjoint and cover the node-set of the given graph. This model is presented in Section 2.

Our goal is not to introduce a model with full mathematical accuracy, rather to build a model that describes all the important requirements and that can be solved efficiently. Of course, those important constraints, that are not included, like maintenance, should be checked. For instance the prescribed maintenance needs to be placed in the task-list of all locomotives used in the transportation, by analyzing (and if it is necessary by modifying) the solution of the $WEAP$. The weak assignment problem is finally presented as a circulation model and we used the minimum cost flow algorithm of LEMON [5]. The computational experiments are provided for the Balassagyarmat region. The results are described in Section 3. Analyzing our computational results, we found out that we can place the maintenance property into the task list of locomotives. So in the region, where we have tested our method, this weak problem seemed to be good for practical purposes. During the computational study we found out that the model is very sensitive for some parameters like connection times, that we denoted by μ. According to the operational policy of MÁV, a train can be pulled by more than one engine (usually by at most two). MÁV allows engines running without a train between two stations, these are said to be light-travels. Such light-travels depend on the given timetable of the region and the applied operational rules and policies. After dispensing with the opportunity of light-travels, other parameters are included in the circulation model of Section 2. In the practice of MÁV the task list of any engine should satisfy a return constraint, namely the engine starts its duty at a given station and after a few days period at the end of a day it arrives at the same station. Such a task list can be repeated periodically. (Of course there are some special days like weekends or holidays when these task lists can not be applied.) Using our model and solution method we can compute such kind of solutions, as well. Taking into consideration that the timetable is periodic we can solve our model on a week (period) base. Periodicity means that on each Monday, before the first train departs, we should have the same number of engines at the departure stations. This condition implies that we have to guarantee the necessary number of locomotives at the beginning of the period. This constraint can be included in the circulation model quite easily without destroying its nice combinatorial properties. In this way we gave a new kind of solution to the engine assignment problem that had not been used at MÁV earlier.

2 Weak Engine Assignment Problem

The integer program presented in [4] describes precisely the engine assignment problem. But this is an NP-complete problem. The condition of maintenance makes the problem NP-complete. The goal of this section is to derive the Weak Engine Assignment Problem (*WEAP*). The *WEAP* does not contain the maintenance conditions, so it is a simplified model of the engine assignment problem. Due to these simplifications the *WEAP* can be solved in polynomial time, based on combinatorial optimization techniques. First we construct a directed acyclic graph, in this graph a path will correspond to a train-sequence, which can be pulled by the same engine. Then we reformulate the problem to a maximum matching problem in a bipartite graph, and finally to a circulation problem. The following notations will be used: the set of arcs leaving (resp. entering) the node v is denoted by $\delta^{out}(v)$ (resp. $\delta^{in}(v)$), the node-set of a graph G is denoted by $V(G)$ and the arc-set is denoted by $A(G)$.

Let us construct a directed graph $G = (V, A)$, where the nodes one-to-one correspond to the trains listed in a given timetable. The train v and the corresponding node is referred also by v. There exists an arc from v_i to v_j, which is denoted by $v_i v_j$ if and only if v_j can be pulled after v_i by the same engine, that is the arrival station of v_i equals the departure station of v_j, and the train v_i arrives at the station at least μ minutes before the train v_j leaves it. Clearly the graph $G = (V, A)$ is a directed acyclic graph.

We refer to the graph $G = (V, A)$ as a *railway graph*, if it comes from *WEAP* by the above construction. A directed path of this graph corresponds to a train-sequence that can be pulled by one engine. (A single-node is also considered as a path.) Each node should be covered by one path. (If we allow only one engine to be coupled to a train, then each node should be covered by exactly one path, in the other case each node should be covered by at least one but at most 2, 3, ...etc. paths.) A set of path $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ is said to be a *disjoint (double, triple) path-cover* if $V(P_i) \cap V(P_j) = \emptyset$ for $1 \leq i < j \leq k$ (or $1 \leq |P_i : v \in P_j| \leq 2, 3$ $\forall v \in V$ resp.), and $\bigcup_{1 \leq i \leq k} V(P_i) = V$. It is easy to see that a disjoint path-cover's composed of t paths corresponds to an engine assignment, which uses t engines. So the *WEAP* can be rephrased as follows:

There is a directed acyclic graph $G = (V, A)$, and we want to find a disjoint path-cover, which has a minimum number of paths. Such a disjoint path-cover gives the minimum number of engines needed, furthermore it gives the train-sequences. Let us construct a bipartite graph $\bar{G} = (\bar{V}^1, \bar{V}^2; \bar{A})$ from the original graph $G = (V, A)$. The node-set $\bar{V}^1$ consists of nodes v^1 for every original node $v \in V$, and the node-set $\bar{V}^2$ consists of nodes v^2 for every original node $v \in V$ likewise. Then the arc-set consists of arcs $v_i^2 v_j^1$ for every original arc $v_i v_j$. A path-cover of the original graph G which has k disjoint paths corresponds to a matching of size $|V| - k$ in the bipartite graph $\bar{G}$. So the minimum disjoint path-cover problem is equivalent to a maximum matching problem in a bipartite graph, which is solvable in polynomial time, see Kőnig [6], or Hopcroft and Karp [7]. But if we want a double or triple path-cover or we have some more conditions, then there is a circulation model that gives us more features.

2.1 Circulation Model

The path-cover is a good model for the *WEAP*, but we have special cases when we need to modify the model. For example if we allow two engines to be coupled on a train. Then the paths, which correspond to the train-sequences, are not necessarily disjoint.

Let us construct the graph $\tilde{G}$ from the railway graph $G = (V, A)$, with node-set $\tilde{V} = V' \cup V'' \cup \{s, t\}$, where V' and V'' are two copies of the original node-set V. Nodes $v', u'' \in \tilde{V} \setminus \{s, t\}$ mean that there exist trains v and u in the given timetable, namely every train is represented twice in $\tilde{G}$. The arc-set is $\tilde{A} = \{v'v'' : v \in V\} \cup \{u''v' : uv \in A\} \cup \{v''t : v \in V\} \cup \{sv' : v \in V\} \cup \{ts\}$, (see Figure 1).

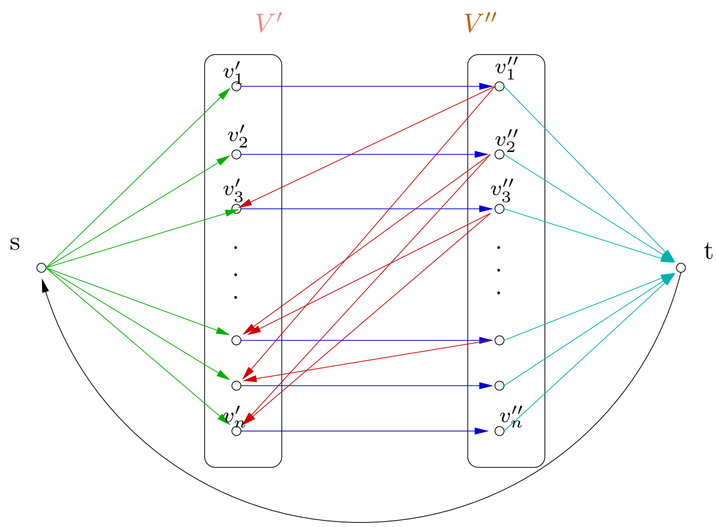

Fig. 1. Circulation

Define the lower bounds and the upper bounds $f, g : \tilde{A} \to \mathbb{N}$ by

$$f(e) = \begin{cases} 1 & \text{if } e = v'v'' \text{ for some } v \in V, \\ 0 & \text{otherwise,} \end{cases} \quad g(e) = \begin{cases} 1 & \text{is } e = v'v'' \text{ for some } v \in V, \\ \infty & \text{otherwise.} \end{cases}$$

A function $x : \tilde{A} \to \mathbb{R}$, for which $f(e) \leq x(e) \leq g(e)$ for every $e \in \tilde{A}$ is said to be a circulation if $\sum_{e \in \delta^{out}(v)} x(e) = \sum_{e \in \delta^{in}(v)} x(e)$ for each $v \in \tilde{V}$. If the upper and lower bounds are integer then the existence of a circulation implies the existence of an integer circulation as in our case. Furthermore the integer circulations correspond to the solutions of the *WEAP*. Moreover, if we define a cost-function $c : \tilde{A} \to \mathbb{R}$ by

$$c(e) = \begin{cases} 1 \text{ if } e = ts, \\ 0 \text{ otherwise.} \end{cases}$$

then the solutions of the *WEAP* correspond to minimum cost circulations.

The previous circulation model describes the *WEAP* completely. The situation when more than one engines are coupled to a train can be described by simple modifications. If for any train v it is allowed to be coupled to at most two engines then $g(e) = 2$ for $e = v'v''$. Similarly we can change the upper bounds for 3 or more engines as well.

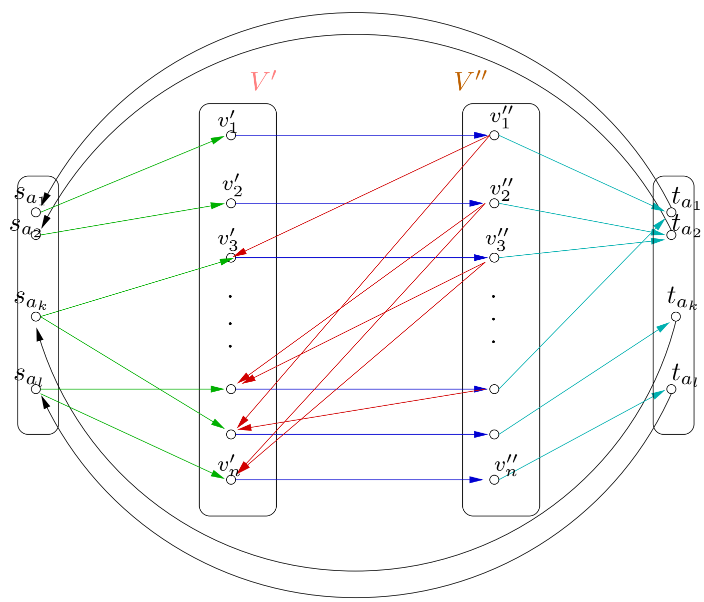

Fig. 2. Circulation for the Periodic Model

For periodic solutions further modification of the above model is needed. The timetable is periodic, so it is a natural aim to look for solutions such that for each station the number of engines is the same at the beginning and at the end of the period. Then let us modify $\tilde{G}$ as follows: replace the nodes s and t by $|N| = l$ pieces of node-pairs, s_a, t_a for each station a. Replace the arc $v't$ by arc $v't_a$ if the arrival-station of the train v' is a, and similarly replace the arc sv'' by arc $s_a v''$ if the departure-station of the train v'' is a, and finally let us put one arc $t_a s_a$ for each $a \in N$. (See Figure 2.)

If we define the lower bounds as zero and the upper bounds as infinity on the new arcs, then the solutions of the circulation problem in this new directed graph correspond to the solutions of the engine assignment problem with periodicity constraint. The periodicity constraint ensures that the same solution can be applied for the next period as well.

The minimum cost circulation problem can be solved in strongly polynomial time, see Goldberg and Tarjan [8], [9].

3 Computational results

We have got data for two regions of Hungary, one is the East-Hungarian Region, and the other is the region of Balassagyarmat. For the East-Hungarian Region we have got the freight-train timetable with 689 trains, 488 trains running with electric engines, and 201 trains running with diesel engines. While these data give two disjoint problems, we considered only the problem, where electric engines are used. The timetable of the freight trains is less dense, so the corresponding railway graph has many disjoint paths. The computations gave a very large number of engines, if we assume that passenger trains are pulled by different sets of engines. This computational experiment shows that there is a strong relation between the minimum number of necessary engines and the density of the railway graph. Certainly in real life the experts of MÁV solve the problem with fewer engines, but the engines pull freight trains and also passenger trains, so the corresponding railway graph is different. If it is necessary then the experts of MÁV define single-engine-runs (light-travels) to make the railway graph more connected. The current procedure of MÁV is purely heuristical.

Let us analyze the currently used timetable of the region of Balassagyarmat. (This timetable is available on http://www.elvira.hu.) In this region we have got a weekly timetable for the passenger trains, with 521 trains and 9 departure/arrival stations. We divided the problem into two parts, with and without the return-back constraint. The program that we have developed controls several parameters like the length of the modeling period (1, 2, 3, 4, 5, 6, 7, 14, 21 days), the connection time, μ, and the maximal number of engines that can be coupled to a train. The starting day of the period can have unexpected effects, for instance the number of necessary engines is different for a one week model if we start the period on Monday or on Tuesday. The tables show the computational results.

Some computational experiences follow: according to the definition of the railway graph, the size of this graph's node-set depends on the length of the period. The size of the edge-set depends on the value of μ, that is if we increase the value of the connection time the graph has less arcs, so the graph would be less dense. Certainly by the growth of the graph we will need more time both to build up the graph, and to solve the problem. The number of engines that can be coupled to a train influences the total number of needed engines. If we increase this number, the number of needed engines will decrease in our case.

The problem with periodicity constraint would become infeasible if we allowed only one engine to be assigned to a train.

Table 1. One-day period without return-condition for the region of Balassagyarmat, when we allow two engines to be assigned to a train

The value of μ	1min	5min	10min	15min	20 min	30min
$\|V(\tilde{G}\|$	81	81	81	81	81	81
$\|A(\tilde{G})\|$	611	602	589	574	567	558
Time to build the graph (s)	0.187	0.141	0.14	0.14	0.141	0.203
Time to solve the problem (s)	0.406	0.437	0.484	0.546	0.547	0.516
Number of needed engines	11	13	14	15	18	19
Number of trains which use 2 engines	2	0	2	2	2	1

Table 2. One-day period with return-condition for the region of Balassagyarmat, when we allow two engines to be assigned to a train

The value of μ	1min	5min	10min	15min	20 min	30min
$\|V(\tilde{G})\|$	81	81	81	81	81	81
$\|A(\tilde{G})\|$	611	602	589	574	567	558
Time to build the graph (s)	0.219	0.188	0.141	0.172	0.156	0.141
Time to solve the problem (s)	0.406	0.484	0.391	0.375	0.375	0.453
Number of needed engines	11	13	14	15	18	19
Number of trains which use 2 engines	2	7	5	5	11	9

Table 3. One-week period without return-condition for the region of Balassagyarmat, when we allow one engine to be assigned to a train

The value of μ	1min	5min	10min	15min	20 min	30min
$\|V(\tilde{G})\|$	521	521	521	521	521	521
$\|A(\tilde{G})\|$	28824	28772	28695	28610	28563	28508
Time to build the graph (s)	6.266	6.281	6.296	5.141	6.296	6.281
Time to solve the problem (s)	10.828	10.25	10.36	12.141	10.297	10.235
Number of needed engines	12	19	21	21	24	25

Table 4. One-week period without return-condition for the region of Balassagyarmat, when we allow two engines to be assigned to a train

The value of μ	1min	5min	10min	15min	20 min	30min
$\|V(\tilde{G})\|$	521	521	521	521	521	521
$\|A(\tilde{G})\|$	28824	28772	28695	28610	28563	28508
Time to build the graph (s)	6.328	6.234	6.25	6.281	6.109	6.203
Time to solve the problem (s)	11.422	10.765	10.844	10.953	10.938	11.016
Number of needed engines	12	13	14	15	18	19
Number of trains which use 2 engines	21	23	38	38	29	23

To solve the problem we use the circulation model (Figure 2) that we extended for different parameters of the problem. (For example the model includes the value of μ, the number of engines that can be coupled to a train, and so on.) The computarial model and solver use extended routines of LEMON. LEMON is an open source library written in C++, developed for combinatorial optimization algorithms by the Department of Operations Research, ELTE [5].

Summarizing our results we can say that a simplified model, *WEAP*, for the engine assignment problem has been introduced. The reason for simplification was that the maintenance constraints make the model NP-complete, therefore it becomes practically non-tractable. Among the advantages of *WEAP* is its pure combinatorial nature and that it is polynomially solvable. On the other hand several operational policies can be built into the circulation model (eg. coupling constraints of engines) keeping its nice optimization property. Our computations show that *WEAP* for small sizes of real life problems can be solved fastly. Further computational investigations are needed for larger problems.

WEAP can be extended for the case when we have several types of trains and several types of engines too. This is ongoing research now.

Acknowledgments

The authors would like to thank Erzsébet Hamvas, Tibor Pólya and József Túri, who work at MÁV Institute for their fruitful discussions and help in collecting the necessary data and analyzing our results. Endre Németh was participating in building the earlier version of the 0-1 programming model. Tibor Takács's cooperation was very helpful in building the communication and data handling parts of our code. This research and the development of the software for *WEAP* was sponsored by the Hungarian Railway Company Pl. (MÁV).

References

1. Ahuja, R., Liu, J., Orlin, J., Sharma, D., Shughart, L.: Solving real-life locomotive scheduling problems. Technical Report 4389-02, Mas-

sachusetts Institute of Technology (MIT), Sloan School of Management (2003) <http://ideas.repec.org/p/mit/sloanp/1799.html> [date of citation: 2006-05-15].
2. Cordeau, J.F., Desaulniers, G., Lingaya, N., Soumis, F., Desrosiers, J.: Simultaneous locomative and car assignment at via rail canada. Transportation Research **Part B** (2000) 767–787
3. Cordeau, J.F., Soumis, F., Desrosiers, J.: Simultaneous assignment of locomotives and cars to passenger trains. Operations Research **49** (2001) 531–548
4. Illés, T., Makai, M., Vaik, Z.: Railway engine assignment models based on combinatorial and integer programing. Technical Report 2005–04, Eővtős Loránd University of Sciences, Department of Operations Research (2005) <http://www.cs.elte.hu/opres/orr/Reports.htm> [date of citation: 2006-05-15].
5. EGRES: Lemon library (2003) <http://lemon.cs.elte.hu> [date of citation: 2006-05-15].
6. Kőnig, D.: Gráfok és mátrixok. Matematikai és Fizikai Lapok (1931) 116–119
7. Hopcroft, J., Karp, R.: A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing (1973) 225–231
8. Goldberg, A.V., Tarjan, R.E.: Finding minimum-cost circulations by canceling negative cycles. Proceeding of the Twentieth Annual ACM Symposium on Theory of Computing, 20th STOC. Chicago, Illinois, 1988 (1988) 388–397
9. Goldberg, A.V., Tarjan, R.E.: Finding minimum-cost circulations by canceling negative cycles. Journal of the Association for Computing Machinery (1989) 873–886