Strategic Gang Scheduling for Railroad Maintenance

Conrado Borraz-Sánchez*, Dengfeng Yang and Diego Klabjan
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208
*Corresponding author’s email: c-borraz-sanchez@northwestern.edu

EXTENDED ABSTRACT

Keywords: Railway Maintenance, Scheduling, Assignment, Integer Programming, Heuristics.

1. INTRODUCTION

The railway industry is nowadays a vital means of transportation worldwide. It owns an intensive infrastructure that requires constant maintenance and improvements. Unlike in Europe, where a track is shared by many railways, in the US, a railway owns the tracks and has to maintain them (typically, tens of thousands of miles of tracks, which are subject to wear and failures due to their extensive use). Such USA’s business model is the focus of this paper.

According to TATA Consultancy Services (2002), the American Public Transportation Association (APTA) estimates that track maintenance expenditures comprise roughly 9% of total operating costs, from which nearly 80% corresponds to labor cost. In monetary terms these numbers may correspond to millions of dollars on annual maintenance projects (Judge, 2010).

The railway maintenance scheduling problem (RMSP) is to find sequences of jobs to be assigned to groups of workers, called gangs, over a time horizon. The planning horizon is typically one year, in which gangs moves from one beat section (a track assigned to a gang to work on) to another upon completion of the work at the section. Several high costs are incurred by such maintenance jobs, including travel allowance and equipment transportation costs. The goal is to minimize the overall cost while obeying a large number of specific business rules (e.g., certain jobs must be performed before other, a gang can only perform a job based on qualifications, job time windows, etc.) usually imposed by railway companies to assure that the maintenance schedule is practically implementable, thus significantly increasing the complexity of this problem.

A Class–1 US railway typically undertakes an intensive effort to determine and sustain annual rail maintenance schedules. In this paper we propose a solution approach to efficiently solve the RMSP. We develop an algorithm that focuses on: 1) minimizing the overall cost incurred by all maintenance jobs within a given planning horizon, including transition costs between jobs, equipment movement costs between two locations, and travel costs between a job location and the home-base of a gang; 2) providing practical feasible solutions that obey all business requirements, for example, gangs are required to work at a southern area during a winter season; and 3) being capable of computationally handling large-scale instances.

Existing approaches either solve the entire problem directly by an optimization solver or iteratively over a subset of gangs. Due to the large number of jobs in our case, none of the existing approaches are applicable to the problem studied herein.

The main contributions of this work are: 1) the design of a solution approach based on very large-scale neighborhood search ideas combined with mathematical programming to solve the RMSP, 2) a new IP model for job re-insertion (the model allows for parts of schedules to be pushed back or forward in time), 3) the efficiency and scalability of the heuristic method, 4) and a real large-scale case study consisting of more than 1,000 jobs and 10–100 gangs over a 365-day planning horizon.

The organisation of this paper is as follows. Section 2 provides the problem statement. Section 3 introduces the job-time network model. Section 4 presents the proposed heuristic approach. The numerical results on a large-scale test case are reported in Section 5. Finally, concluding remarks are given in Section 6. The related work is discussed next.
1.1. Related Work

A survey of the literature reveals that not much relevant prior work has been reported in this area. Relevant works are due to Gorman and Kanet (2010), Gang (2006), and Peng et al. (2011). Gorman and Kanet (2010) proposed and provided a comparison between a time-space network model and a job-scheduling network model to solve the RMSP. Due to the considerably large size of our instance, the models in Gorman and Kanet (2010) can not be applied to our case, i.e., a solver can not directly solve our instance and business requirements.

Gang (2006) proposed a preprocessing procedure to significantly reduce the size of the gang scheduling model of an unnamed railway. The preprocessing technique basically reduces the size of the repositioning network and job time windows by combining jobs and eliminating variables and constraints. Consequently, a commercially available solver, CPLEX, is applied to a series of simpler problems, which are extensions of the original problem. We differ from Gang’s work in that we tackle the complete model with a mathematical programming-based heuristic.

More recently, Peng et al. (2011) propose a heuristic solution approach to solve the RMSP. The solution procedure is based on an iterative heuristic algorithm that is applied to a time-space network flow model. Contrary to our work, the heuristic method iteratively solves a sequence of subproblems with no more than two gangs each. Given the relatively small size of the resulting subproblems, they use a commercial optimizer (CPLEX) to solve them, thus imposing severe limitations to the search space. In our case-study, even a model with two gangs can not be solved by a solver due to the sheer size of the problem.

Unlike in our proposed improvement stage, in which diverse analytical methods are applied, the solution improvement stage presented by Peng et al. (2011) is based on a random project swapping procedure that is called inside a loop. Due to the significant computational resources required by their heuristic method, coupled with the selection of two gangs to create the IP model, they prioritize the subproblems in such a way that a more promising improvement may be accomplished. The case-study reported in their work contains a total of 333 jobs to be assigned to 21 gangs within a 48-week planning horizon. This results in a significantly smaller test case than the one solved in our work.

To summarize, Gorman and Kanet (2010) and Gang (2006) solve models directly by optimization solvers, which is not applicable to our instance due to the large number of gangs and jobs. Peng et al. (2011) tackle the problem iteratively. Their methodology includes a step that does not scale to the problem of our size and thus their methodology is also not applicable.

2. Problem Statement

The RMSP poses a complex optimization task that results in a combined job assignment-sequencing-timetabling problem. Given a set of gangs and jobs, whereby each gang can perform a job in a given amount of time and at a given cost, assign each job to a gang and timetable them to minimize total cost.

A typical major Class–1 US railway deploys 4 types of gangs: rail, tie and surface (T&S), curve, and dual, which have to be scheduled over a span of one year. This results in a total of 10-100 gangs and several thousand jobs around the U.S. Moreover, each gang works on a set of jobs that are mutually independent of other gangs, with each job being specified by attributes (earliest start date, latest start date, duration, division, location, etc.). In addition, certain jobs must obey precedence constraints, e.g., work on ties must precede any work on the rails. From the operational perspective, it is not desirable to perform two jobs geographically close to each other during the same period since it could substantially disrupt train operations. The fewer occurrences of such situations, the more robust the schedule becomes.
Algorithm 1 – Solving RMSP by Alg1

**Input:** An instance of the RMSP.

Step 1: Construct the job-time network for each gang type.
Step 2: Find a path for each gang by running a DP-based shortest path algorithm.
Step 3: Phase 1: Find an initial solution by starting from solution obtained in Step 2.
Step 4: Phase 2: Apply an enhanced solution procedure to the initial point found in Phase 1.

**Output:** A favorable job-gang schedule for the RMSP.

3. NETWORK MODEL

A key observation when designing the network model is the existence of a unique set of exclusive jobs that can be assigned to each gang type. Individual job-gang networks are hence required. Note, however, that the presence of job precedence rules makes the problem inseparable. The networks are linked to each other by a supersource node and a supersink node, which have virtually unlimited supply and demand of gangs, respectively.

Let \( G_v \) be the set of gang instances of type \( v \). Then, \( \mathcal{G}^g = (N^g, A^g) \) represents a network instance of type \( g \in G_v \), where \( N^g \) and \( A^g \) are the node and arc sets, respectively. The network is composed of daily time as x-coordinate and job as y-coordinate. Each vertex or node \( (j, t) \) represents the available start time \( t \) (in daily increments) for each job \( j \) that can be processed by gang \( g \). Let \( P^g \) be the set of paths (job node sequences) for gang \( g \) starting at job \( s^g \), with an associated cost \( c^g_p \).

4. SOLUTION METHODOLOGY

We propose a two-phase heuristic approach for the RMSP, based on very large-scale neighborhood search ideas and mathematical programming. We start by constructing a job-time network model for each gang type and a set of gang-paths by applying a shortest path algorithm using a dynamic programming technique. Next, the set of job-gang paths is used to find an initial point in Phase 1. Then, a mathematical programming-based solution approach is applied in Phase 2 to improve the initial solution. A summary of the solution framework is given in Algorithm 1.

Due to the strong requirements imposed by the union and regulations, the gang paths provided by the shortest path algorithm in Step 2 of Algorithm 1 may not cover all jobs of the planning horizon, thus leading to a pool of unassigned jobs. Here, Phase 1 is then executed to find an initial feasible solution.

An iteration of Phase 1 basically consists of one insertion step and one swap step. The former step largely decreases the number of unassigned jobs by inserting them in the existing paths and pushing back the subsequent jobs in the paths. However, the limited flexibility of the available paths due to the tight time windows of their jobs may lead to violations when pushing back jobs in the paths. Here, several ‘unassigned’ jobs may remain unassigned. Consequently, the swap and increasing flexibility procedures are called to enforce that all unassigned jobs are scheduled. The basic idea behind this step is to swap the unassigned jobs and those jobs in the paths while interfering with the smallest number of possible jobs of the current path.

Once a feasible initial solution is obtained, Phase 2 is then executed to iteratively improve it. First, we randomly extract short sequences of assigned jobs from the paths (see Fig. 1). This is accomplished by selecting specific points that are strategically spanned over the whole path. By doing so, we create enough available space for the new sequences of job nodes. Second, we recombine all extracted/uncovered nodes into a pool of short subsequences and reinsert them back into the short-cut solution. Next, we formulate and solve a reinsertion ILP model to optimally create subsequences of unassigned jobs (nodes) for each path. Finally, we incorporate the optimal IP-solution provided by the reallocation ILP model into the current solution. This
4

reinsertion step marks the end of an iteration and the whole procedure is repeated until an iteration limit is reached or the solution is significantly improved.

Fig. 1. Pool of extracted/uncovered jobs for recombination.

5. Numerical Results

The computational results reported herein correspond to a large-scale real-world test case consisting of more than 1,000 jobs and 30 gangs with 4 gang types over a 365-day planning horizon. Data, while being supplied by a major Class–1 US railway, are treated with confidentiality and their disclosure is somehow limited. The solution procedures were developed in Visual Studio C++ v.2008 and run on a 2.4 GHz Intel(R) processor with 2 GByte RAM under MS-Windows operating system.

Here, we want (1) to evaluate the quality of the gang schedule provided by Algorithm 1 by comparing it to the original schedule performed by the current industry practice, and (2) to assess the performance of our heuristic method.

Table 1 shows the comparison analysis between 3 Alg1-solutions and the baseline-solution used by the railway. The only difference among the three solutions is the weight between the equipment movement cost and the remaining cost components. In the table, the total cost, travel allowance, equipment movement (EM) cost and robustness are shown for each solution shown in Column 1. The values correspond to the relative improvement of Alg1-solutions over baseline-solution. Soft violations are included in the last column. Note that the robustness of the solution is nothing but the number of pairs of jobs that are within a certain distance to each other and are performed in overlapping times.

Table 2 – Sensitivity analysis on the robustness of the solution.

From the results shown in Table 1, we can make three key observations. First, all Alg1-solutions outperform the baseline-solution with a relative improvement up to 18.8%, which may result, in monetary terms, in millions of dollars of annual saving for the railway industry. Second, soft precedence rule violations were reduced by 100%. Third, Alg1-solution B achieved the best robustness, yet its total cost is outperformed by Alg1-solution A. Here, a trade off between total cost and robustness may become an important factor for managers of the railway industry.
Table 2 shows the outcome of conducting a sensitivity analysis to evaluate the effect of the robustness penalty on total cost values. In the table, we observe that the best solution achieved by our heuristic was provided by a robustness penalty set to 1500. Here, both total costs and robustness violations were improved above 14% and 30%, respectively, when compared to the baseline-solution.

Fig. 2 shows the performance of Algorithm 1. From the figure, we can observe that a significant decrease in the total cost is obtained by Phase-1 (initial solution) when compared to the baseline solution. Similarly, our initial solution is significantly improved by the first iteration of Algorithm 1’s Phase-2. We observe that total costs keep decreasing until no significant improvement is achieved after 8 iterations. Concerning the running time, Phase-1 required roughly 1.0 CPU-time hour to provide an initial solution, whereas Phase-2 finished after 8 iterations in about 1.5 CPU-time hours. This proves the capability of our heuristic approach when solving large-scale test cases of the RMSP.

6. CONCLUSIONS

We have proposed and developed a totally new and very suitable solution approach for the RMSP. Our solution methodology applies inside a loop a heuristic method that overcomes cycle problems by using different penalty functions.

Several computational experiments were conducted to show the capability of our heuristic. The results showed a relative improvement above 14% of our gang schedule when compared to the baseline schedule performed by the current industry practice. This is a very favorable result since it significantly reduced the overall maintenance cost for the railway industry.

As a final remark, our heuristic proved to be as effective as efficient while requiring about 2.5 CPU-time hours to schedule more than 1,000 maintenance projects with time windows under the constraints of job precedence and robustness rules. This is reasonable for the railway managers to find favorable gang maintenance schedules.

REFERENCES


