Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 58 (2017) 215–222 www.elsevier.com/locate/endm

Scheduling preventive railway maintenance activities with resource constraints

Rita Macedo [a,1] Rachid Benmansour [b,2] Abdelhakim Artiba [b,3]
Nenad Mladenović [b,4] Dragan Urošević [c,5]

[a] *Univ. Lille, CNRS, UMR 9221 - LEM Lille Économie Management, F-59000 Lille, France*

[b] *University of Valenciennes and Hainaut Cambrésis, France*

[c] *Mathematical Institute, Serbian Academy of Science and Arts, Belgrade, Serbia*

Abstract

In this paper, we focus on the scheduling of preventive railway maintenance activities. The objective is to keep the railway infrastructure in good operating conditions at low costs, also taking into account the limited available resources in what concerns crew members. Equipments degrade with usage and age and a good preventive maintenance program can greatly reduce their unreliability in the sense that expectable failures can be anticipated. We propose a mixed integer programming formulation for the problem of scheduling preventive railway maintenance activities and a Variable Neighborhood Search (VNS) algorithm to solve large instances of the problem.

Keywords: Variable neighborhood search, scheduling, maintenance, railway.

[1] Email: rita.macedo@univ-lille3.fr
[2] Email: rachid.benmansour@univ-valenciennes.fr
[3] Email: abdelhakim.artiba@univ-valenciennes.fr
[4] Email: nenad.mladenovic@univ-valenciennes.fr
[5] Email: draganu@turing.mi.sanu.ac.rs

http://dx.doi.org/10.1016/j.endm.2017.03.028

1 Introduction

Rail transport is one of the safest and most environmentally friendly means of conveyance of passengers and goods. By making regions and markets more accessible, it plays a main role in the development of countries due not only to its impact on the economy but also to its social role. In order to support the increase of traffic due to globalization and personal interchanges, many efforts have to be done to keep rail transport safe, efficient and competitive. This can be achieved through technical elements like supervision, maintenance, and standardization.

In this paper, we focus on the scheduling of preventive railway maintenance activities. The objective is to keep the railway infrastructure in good operating conditions at low costs, also taking into account the limited available resources in what concerns crew members.

Equipments degrade with usage and age and a good preventive maintenance program can greatly reduce their unreliability in the sense that expectable failures can be anticipated. Although this represents additional costs, these are typically much less important than the ones caused by the failure of an equipment, which can cause the failure of a complete system, added of a subsequent corrective maintenance. In sum, to prevent that stochastic failures occur frequently on the railway infrastructure, it is important to perform preventive maintenance on a regular basis. This helps to reduce the probability of the occurrence of a failure on the components of the railway infrastructure and maximizes the operational benefits [5,1]. The main goal of preventive maintenance is to prevent possible failures before they actually happen, reducing costs and increasing reliability of equipments and services.

We propose a mixed integer programming formulation for the problem of scheduling preventive railway maintenance activities and a Variable Neighborhood Search (VNS) algorithm to solve large instances of the problem.

2 Problem description

There is a set of maintenance activities to perform during a planning horizon composed of $|T|$ periods. As pointed out by [3], maintenance activities in railways can be divided into two categories: small and large routine works. Therefore, we consider two different kinds of maintenance activities: routine works with smaller durations and projects with larger durations. Routine works, such as inspections, cleaning operations and small repairs, are conducted on a periodic basis, whereas projects are considered to be conducted

once within the horizon (ballast cleaning, rail grinding, etc.). More formally, the maintenance activities to be performed belong to one of two different sets: routine maintenance activities (R) or projects (P). The set of all activities is therefore defined as $A = R \cup P$. Routine maintenance activities are performed at a single period and are cyclic. Each activity $a \in R$ has a defined frequency $F^a = \left\lfloor \frac{|T|}{L^a} \right\rfloor$, where L^a denotes the duration of the interval between two consecutive repetitions of activity a. Projects are performed only once but have a duration D_p that is typically larger than one period. Each project $p \in P$ must begin at a period belonging to a defined interval T_p and its activities continue to be performed during the $D_p - 1$ subsequent periods.

Whenever at least one activity is being performed at a period $t \in T$, the rail link must be closed and the system incurs into a holding cost c_t. This cost is independent of the number of activities being performed. This means that it is interesting to try to combine activities to be allocated to the same periods.

However, there are some activities that are incompatible. A given set $C = \{(a_1, a_2) |$ activities $a_1, a_2 \in A$ can be performed at the same period$\}$ defines the compatible activities, i.e. the pairs of activities that can be performed at a same period. This problem was first described in Budai et al. [2], where the authors proposed a Mixed Integer Programming (MIP) formulation and several heuristics to solve the problem.

We further consider that there is a resource limitation. It is related to the fact that performing activities implies having a sufficiently large number of crew members and equipments. It is defined that the capacity of the company in what concerns these two resources is of performing at most θ activities at the same period. Each additional activity incurs into a penalization α_t that may represent additional costs of outsourcing, paying extra hours to workers, renting additional machines, among others.

3 Mathematical formulation

Let x_t^a, y_t^p, m_t be binary variables defining respectively whether activity $a \in A$ is performed at period $t \in T$ or not, whether project $p \in P$ is started at period t or not and whether there is any activity being performed at instant $t \in T$ or not. The difference between the sum of allocated activities at a given period $t \in T$ and the maximum desirable number of activities θ is represented by the integer variable δ_t.

This problem can be modeled with the following MIP formulation with

assignment and positional date variables.

$$\min \sum_{t\in T} c_t m_t + \sum_{t\in T} \alpha_t \delta_t \tag{1}$$

$$s.t. \sum_{t=1}^{L^a} x_t^a = 1 \quad \forall a \in R \tag{2}$$

$$x_t^a = x_{t+qL^a}^a \quad \forall a \in R, t \in \{1,\dots,L^a\}, 1 \le q \le F^a - 1 \tag{3}$$

$$\sum_{t\in T_p} y_t^p = 1 \quad \forall p \in P, \tag{4}$$

$$x_s^p \ge y_t^p \quad \forall p \in P, t \in T_p, s = t,\dots,t + D_p - 1, \tag{5}$$

$$x_t^m + x_t^n \le 1 \quad \forall t \in T, (m,n) \notin C, \tag{6}$$

$$m_t \ge x_t^a \quad \forall t \in T, \forall a \in A, \tag{7}$$

$$\delta_t \ge \sum_{a\in A} x_t^a - \theta \quad \forall t \in T, \tag{8}$$

$$\delta_t \ge 0 \quad \forall t \in T, \tag{9}$$

$$x_t^a, y_t^p, m_t \in \{0,1\} \quad \forall t \in T, \forall a \in A, p \in P \tag{10}$$

The objective function (1) minimizes the total operational costs, which comprise the track occupancy costs and the penalization costs of assigning more than θ activities to the same period. The unit penalization cost of period t is denoted by α_t. Every cyclic routine maintenance activity $a \in R$ must be performed at regular intervals of L^a periods (constraints (2) and (3)), and every project must begin at a period within its beginning interval T_p and continue through the $D_p - 1$ subsequent periods (constraints (4) and (5)). Constraints (6) ensure that only compatible activities are performed at the same period and constraints (7) guarantee that a track occupancy cost will be taken into account for every period with at least one allocated activity. Finally, constraints (8) and (9) define the number of activities that surpass θ for every time period.

4 Variable Neighborhood Search algorithm

We propose a Variable Neighborhood Search (VNS) [6,4]for this problem. A solution S is represented by an array $S = \{s_a | a \in A\}$, $|S| = |A|$, where s_a represents the first period where a is performed, if a is a routine maintenance activity, or the starting period of a, if a is a project. Therefore, $1 \le s_a \le$

$L^a, \forall a \in R$, and $s_a \in T_a, \forall a \in P$.

A solution may be infeasible if there is any pair of incompatible activities being performed at the same period. Thus, the objective function for a solution S is defined as:

$$f(S) = \sum_{t \in T} c_t m_t(S) + \sum_{t \in T} \alpha_t \delta_t + P_f \times NbConf(S),$$

where $m_t(S)$ is equal to 1 if there is any activity being performed at period t and equal to 0 otherwise, P_f is a penalty factor and $NbConf(S)$ is the number of pairs of conflicts in solution S.

As explained before, δ_t represents the difference between the sum of allocated activities at a given period $t \in T$ and the maximum desirable number of activities θ, i.e. $\delta_t = na_t - \theta$, where na_t represents the number of activities allocated to period t. We compute na_t as

$$na_t = |\{a \in P : s_a \leq t < s_a + D_a\}| + |\{a \in R : L^a | (t - s_a)\}|$$

and m_t as

$$m_t = \begin{cases} 0, \text{ if } na_t = 0; \\ 1, \text{ if } na_t > 0. \end{cases}$$

The initial solution (`Initial_solution`) is random. Each $s_a \in S$ corresponds to a random integer number belonging to $[1, L^a]$ if $a \in R$ and to T_a if $a \in P$.

The shaking (`Shake`) is performed in neighborhoods N_k and consists of k randomly selected moves. Each move consists of changing s_a for randomly selected activity $a \in A$, being the new value of s_a also determined at random.

For the Local Search (`LocalSearch`), we consider a single type of move in a neighborhood consisting of all solutions obtained a value s_a to one of its allowed values, belonging to $[1, L^a] \setminus \{s_a\}$ if a is a routine maintenance activity or $T_a \setminus \{s_a\}$ if a is a project.

The method is described in Algorithm 1.

Algorithm 1 VNS algorithm

Function VNS $(k_{min}, k_{max}, k_{step}, t_{max})$
$S \leftarrow$ `Initial_solution`
$k \leftarrow k_{min}$
repeat
    $S' \leftarrow$ `Shake`(S, k)
    $S'' \leftarrow$ `LocalSearch`(S')
    if $f(S'') < f(S)$ **then**
        $S \leftarrow S''$
        $k \leftarrow k_{min}$
    else
        $k \leftarrow k + k_{step}$
        if $k > k_{max}$ **then**
            $k \leftarrow k_{min}$
        end if
    end if
until $CPUtime() > t_{max}$
return S

5 Computational Results

We conducted computational experiments on a set of 30 instances adapted from [2], with 15, 20 or 25 routine activities, a number of projects between 0 and 2, $c_t = 25$, $\forall t \in T$, $\theta = 3$ and $\alpha_t = 30$, $\forall t \in T$.

Table 1 reports the computational results obtained with VNS (columns *VNS*), compared with the ones obtained by solving model (1)-(10) with CPLEX (columns *MIP*). Both methods were run within a time limit (t_{max}) of 1800 seconds.

Columns 2-5, 6-9 and 10-13 describe, respectively, the results for instances with 15 ($|R| = 15$), 20 ($|R| = 20$) and 25 ($|R| = 25$) routine activities. Columns *obj* report the best found solution and columns t its corresponding computational time.

Table 1: Computational Results

	$\|R\| = 15$				$\|R\| = 20$				$\|R\| = 25$			
	MIP		*VNS*		*MIP*		*VNS*		*MIP*		*VNS*	
	obj	*t*	*obj*	*t*	*obj*	*t*	*obj*	*t*	*obj*	*t*	*obj*	*t*
1	2545	817,82	2545	0,04	3380	1194,69	3380	2,10	4610	85,57	4610	2,54
2	2275	422,38	2275	1,77	2685	187,53	2685	3,57	3680	1800,02	**3650**	0,60
3	2615	1089,44	2615	2,85	2470	368,48	2470	0,93	3560	1800,01	3560	0,02
4	2555	277,75	2555	0,04	3170	715,77	3170	0,11	4880	767,44	4880	0,03
5	2100	234,03	2100	0,25	2645	1800,01	2645	1,16	3470	1800,01	**3440**	3,01
6	2575	130,38	2575	0,03	2690	1800,01	2690	0,12	3920	1800,01	**3860**	0,49
7	2590	535,36	2590	1,20	3350	1800,01	**3320**	0,10	3950	1 800,01	3950	1,58
8	2325	1800,01	2325	2,85	3740	988,40	3740	2,70	3470	1800,01	**3380**	2,85
9	1850	479,54	1850	2,35	2425	1800,02	**2390**	2,16	6050	10,32	6050	1,30
10	2765	279,82	2765	0,35	2870	1800,01	2870	-	5480	1014,74	5480	0,01

CPLEX solved 60% of the instances to optimality, within the time limit. For those instances, VNS always found the optimal solution. CPLEX did not prove the optimality of the solutions of 12 instances. For 6 of them, VNS found the same solution and for the other 6 it found better solutions (marked in bold in Table 1). CPLEX took an average computational time of 1047.7 seconds, while VNS only took on average 1.2 seconds to reach the best found solutions.

6 Conclusions

We address a scheduling preventive railway maintenance activities problem and propose a Mixed Integer Programming (MIP) formulation and a Variable Neighborhood Search (VNS) algorithm to solve this problem.

Computational results show that our VNS is quite efficient for the tested instances, always providing the optimal solutions for the instances where the MIP model was able to find them.

References

[1] Benmansour, R., H. Allaoui, A. Artiba and S. Hanafi, *Minimizing the weighted sum of maximum earliness and maximum tardiness costs on a single machine with periodic preventive maintenance*, Computers & Operations Research **47** (2014), pp. 106–113.

[2] Budai, G., D. Huisman and R. Dekker, *Scheduling preventive railway maintenance activities*, Journal of the Operational Research Society **57** (2006), pp. 1035–1044.

[3] Esveld, C., "Modern railway track," MRT-productions Zaltbommel, The Netherlands, 2001.

[4] Hansen, P., N. Mladenović and J. A. M. Pérez, *Variable neighbourhood search: methods and applications*, Annals of Operations Research **175** (2010), pp. 367–407.

[5] Kumar, U. D., J. Crocker, J. Knezevic and M. El-Haram, "Reliability, Maintenance and Logistic Support:-A Life Cycle Approach," Springer Science & Business Media, 2012.

[6] Mladenović, N. and P. Hansen, *Variable neighborhood search*, Computers & Operations Research **24** (1997), pp. 1097–1100.