

# Maintenance Scheduling in a Railway Corridor

**Saman Eskandarzadeh<sup>a</sup>, Thomas Kalinowski<sup>a</sup> and Hamish Waterer<sup>a</sup>**

<sup>a</sup>*School of Mathematical and Physical Sciences, University of Newcastle, Callaghan, NSW 2308*  
Email: [saman.eskandarzadeh@newcastle.edu.au](mailto:saman.eskandarzadeh@newcastle.edu.au)

**Abstract:** Australia has a large operational heavy railway network which is approximately 33,355 route-kilometres. This network accounted for approximately 55 percent of all freight transport activity in Australia in the financial year 2013-14, almost 367 billion tonne-kilometres which was up 50 percent from 2011-12 (BITRE (2016)).

To provide a safe and reliable railway network to customers, an effective maintenance regime is a key requirement. Planned maintenance and asset renewal in capital intensive industries such as the railway industry, which has expensive infrastructure, is a common and effective maintenance practice. Infrastructure is of essential importance to maintain a reliable customer service. To prevent long unplanned interruptions in the service to customers, a proper maintenance and renewal program is required. The objective is to schedule planned maintenance and asset renewal jobs in such a way that their impact on the capacity that will be provided to customers is minimised while at the same time keeping the infrastructure in good working condition. A proper maintenance and renewal schedule limits the frequency with which disruptive emergency maintenance is needed.

We investigate a planned maintenance and asset renewal scheduling problem on a railway corridor with train traffic in both directions. Potential train journeys are represented by train paths, where a train path is specified by a sequence of (location,time)-pairs, and we distinguish between up- and down paths, depending on the direction of travel. Necessary maintenance and renewal activities, or work, are specified by a release time, a deadline, a processing time and a location. Scheduling work at a particular time has the consequence that the train paths passing through the corresponding location while the work is carried out have to be cancelled. An instance of the problem is given by a set of train paths and a set of work activities, and the task is to schedule all the work such that the total number of cancelled paths is minimised.

There is a vast literature on scheduling problems and on transportation network problems. However, the interaction of these problems in contexts such as the railway industry have not been studied thoroughly. Boland et al. (2014) study the problem of scheduling maintenance jobs in a network. Each maintenance job causes a loss in the capacity of network while it is being done. The objective is to minimize this loss or equivalently maximize the capacity over time horizon in such a way that all jobs are scheduled. They model the problem as a network flow problem over time. This problem and its variants are investigated in Boland et al. (2013), Boland et al. (2014), Boland et al. (2015), Boland et al. (2016) and Abed et al. (2017). Our work is different from the previous works. The main difference is that we model the capacity by train paths whereas in a network flow model over time the capacity is modelled approximately by flows over time.

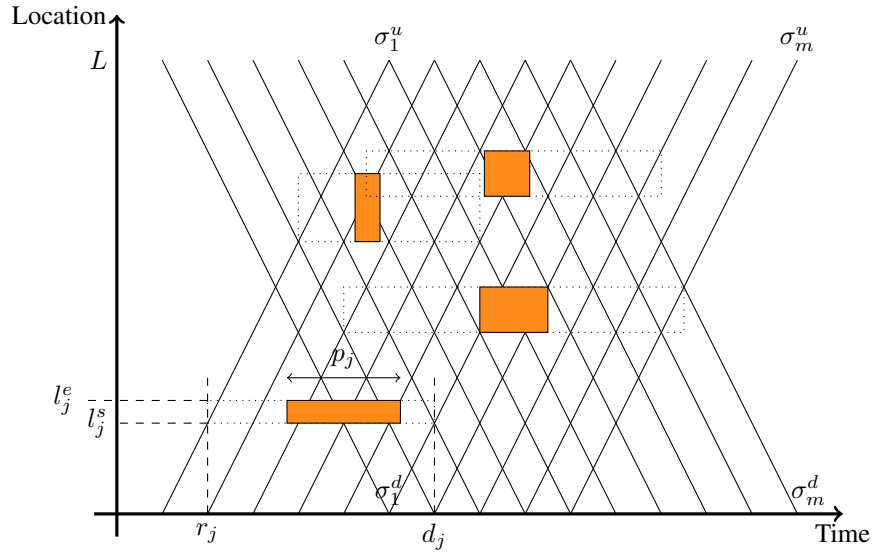
The primary purpose of this study is to provide further insight into this problem, to characterize the structural properties of optimal solutions, and to use these properties to develop efficient combinatorial and integer programming based solution algorithms. We present theoretical and computational results on the performance of the developed solution approaches.

**Keywords:** *Maintenance scheduling, combinatorial optimization, integer programming, computational complexity*

## 1 PROBLEM DESCRIPTION

We are given a set of maintenance jobs  $\mathcal{J}$ , and a set of paths  $\mathcal{P}$ . Each job  $j \in \mathcal{J}$  is specified by its earliest start time  $r_j$ , its latest finish time  $d_j$ , its processing time  $p_j$ , its start location  $l_j^s$  and its end location  $l_j^e$ , all of which are nonnegative real numbers. There are train paths in two opposite directions. Depending on the direction of a path, it is called either up-path or down-path. Let  $[(x_1, y_1), (x_2, y_2)]$  denote a line segment connecting two points  $(x_1, y_1) \in \mathbb{R}^2$  and  $(x_2, y_2) \in \mathbb{R}^2$ . Let  $\Delta, \delta, L$  be positive real numbers. The set of up-paths is denoted by  $\mathcal{P}^u = \{\sigma_1^u, \dots, \sigma_m^u\}$  where  $\sigma_i^u = [(i\Delta, 0), (i\Delta + \delta, L)]$  and the set of down-paths is denoted by  $\mathcal{P}^d = \{\sigma_1^d, \dots, \sigma_m^d\}$  where  $\sigma_i^d = [(i\Delta, L), (i\Delta + \delta, 0)]$ . The parameters  $\Delta, \delta$  and  $L$  are the headway time, the travel time, and the length of the corridor, respectively. If a location is possessed by a job in certain time interval, then any path that passes this location during this time has to be cancelled. Note that different jobs can possess the same location at the same time. The decision in the maintenance scheduling problem is to find start times of jobs in such a way that the number of cancelled paths is minimised. A solution is given by a vector  $s = (s_j)_{j \in \mathcal{J}}$  of start times.

An instance of the problem can be represented geometrically in the plane where the horizontal and vertical axes represent time and location, respectively. Paths correspond to line segments and jobs to rectangular boxes. An instance of the problem with four jobs is depicted in Figure 1.



**Figure 1.** An instance of the maintenance scheduling problem with four jobs. The dotted boxes represent the jobs, and the filled boxes indicate a feasible solution. The set of cancelled paths is

$$\{\sigma_1^d, \sigma_2^d, \sigma_4^d, \sigma_6^d, \sigma_7^d, \sigma_8^d, \sigma_2^u, \sigma_3^u, \sigma_4^u, \sigma_5^u, \sigma_6^u, \sigma_7^u\}.$$

Corresponding to starting job  $j \in \mathcal{J}$  at time  $s_j \in [r_j, d_j - p_j]$ , there is a set of paths that have to be cancelled due to overlap with that placement. For each job  $j \in \mathcal{J}$ , let  $\mathcal{R}_j$  denote the collection of sets of cancelled paths corresponding to feasible start times for job  $j$ . We call the path sets  $R \in \mathcal{R}_j$  *possessions*. The problem of finding an optimal start time vector  $(s_j)$  is equivalent to selecting possessions  $R_j \in \mathcal{R}_j$  such that the cardinality of their union is minimised. If  $\emptyset \in \mathcal{R}_j$  then the  $j$  can be removed from  $\mathcal{J}$  without changing the problem. If  $R \subseteq R'$  for two elements  $R, R' \in \mathcal{R}_j$  then  $R'$  can be removed from  $\mathcal{R}_j$ , because in any optimal solution  $R'$  can be replaced by  $R$ . Finally, if there are two jobs  $j, j'$  such that for every  $R \in \mathcal{R}_j$  there exists an  $R' \in \mathcal{R}_{j'}$  with  $R' \subseteq R$  then  $j'$  can be removed from  $\mathcal{J}$  without changing the optimal objective value. In summary, we can make the following assumptions:

- For all  $j \in \mathcal{J}$ ,  $\emptyset \notin \mathcal{R}_j$ .
- For all  $j \in \mathcal{J}$  and distinct  $R, R' \in \mathcal{R}_j$ ,  $R \not\subseteq R'$ .
- For all  $j, j' \in \mathcal{J}$ , there exists  $R \in \mathcal{R}_j$  such that  $R' \not\subseteq R$  for all  $R' \in \mathcal{R}_{j'}$ .

In this paper, due to space constraints, we omit the proofs. The proofs are included in Eskandarzadeh et al. (2017)

## 2 MIP FORMULATIONS

For every path  $\sigma \in \mathcal{P}$ , let  $y_\sigma$  be a binary variable which takes value one if and only if path  $\sigma$  is cancelled. For every  $R \in \mathcal{R} = \cup_{j \in \mathcal{J}} \mathcal{R}_j$ , let  $x_R$  be a binary variable which takes value one if at least one job is scheduled which overlaps exactly the paths in  $R$ . Now, we can formulate the problem as follows:

$$\begin{aligned}
 \text{[HSM]} \quad & \text{Minimise } \sum_{\sigma \in \mathcal{P}} y_\sigma \text{ subject to} \\
 & \sum_{R \in \mathcal{R}_j} x_R \geq 1 && \text{for all } j \in \mathcal{J}, \\
 & y_\sigma \geq x_R && \text{for all } R \in \mathcal{R}, \sigma \in R, \\
 & x_R \in \{0, 1\} && \text{for all } R \in \mathcal{R}, \\
 & y_\sigma \in \{0, 1\} && \text{for all } \sigma \in \mathcal{P}.
 \end{aligned}$$

In the following, we present another Mixed-integer programming (MIP) formulation for the case that there are only up-paths, i.e.,  $\mathcal{P}^d = \emptyset$ . We drop the upper index  $u$  and denote the paths by  $\sigma_i$ .

**Definition 1.** We call the set of consecutive paths from path  $\sigma_i$  to path  $\sigma_k$  a *span* from path  $\sigma_i$  to path  $\sigma_k$  and denote it by  $[\sigma_i, \sigma_k]$ .

A feasible solution for the maintenance scheduling problem can be interpreted as a collection of disjoint cancelled spans of paths. More formally, a collection  $([\sigma_i, \sigma_k])_{(i,k) \in B}$  of disjoint spans represents a feasible solution if for every  $j \in \mathcal{J}$  there exists an  $R_j \in \mathcal{R}_j$  such that

$$\bigcup_{j \in \mathcal{J}} R_j \subseteq \bigcup_{(i,k) \in B} [\sigma_i, \sigma_k].$$

Now, we can formulate maintenance scheduling problem with unidirectional traffic as a set covering problem. For  $1 \leq i \leq k \leq m$ , let  $x_{ik}$  be a binary variable indicating that the paths in  $[\sigma_i, \sigma_k]$  are cancelled. Let  $F_0 = \{(i, k) : 1 \leq i \leq k \leq m\}$ , and for  $j \in \mathcal{J}$ , let  $B_j = \{(i, k) \in F_0 : \exists R \in \mathcal{R}_j, R \subseteq [\sigma_i, \sigma_k]\}$ . Then the formulation is

$$\begin{aligned}
 \text{[SCM]} \quad & \text{Minimise } \sum_{(i,k) \in F_0} (k - i + 1)x_{ik} \text{ subject to} \\
 & \sum_{(i,k) \in B_j} x_{ik} \geq 1 && \forall j \in \mathcal{J}, \\
 & x_{ik} \in \{0, 1\} && \forall (i, k) \in F_0.
 \end{aligned} \tag{1}$$

Note that we can replace  $F_0$  by the set

$$F = \{(i, k) \in F_0 : \exists j_1, j_2, \dots, j_r \in \mathcal{J}, R_{j_1} \in \mathcal{R}_{j_1}, \dots, R_{j_r} \in \mathcal{R}_{j_r}, \bigcup_{t \in [1, r]} R_{j_t} = [\sigma_i, \sigma_k]\}.$$

Let  $r'_j, d'_j$  be the index of the leftmost and rightmost paths that are in the set  $\cup_{R \in \mathcal{R}_j} R$ .

**Definition 2.** An instance of the maintenance scheduling problem is said to have interval property if for every  $j \in \mathcal{J}$ , the elements of  $\mathcal{R}_j$  have the same cardinality, say  $p'_j$ . In other words,  $\mathcal{R}_j$  has the form

$$\mathcal{R}_j = \{[\sigma_{r'_j}, \sigma_{r'_j+p'_j-1}], [\sigma_{r'_j+1}, \sigma_{r'_j+p'_j}], \dots, [\sigma_{d'_j-p'_j+1}, \sigma_{d'_j}]\}.$$

In the rest of this section, we consider instances of maintenance scheduling problem that have interval property and include only up-paths.

**Lemma 1.** Suppose the number of cancelled paths is the same for all jobs, say  $p'_j = p'$  for all  $j \in \mathcal{J}$ . Then the polytope defined by (1) and  $0 \leq x_{ik} \leq 1$  for all  $(i, k) \in F$  is integral.

This follows from the observation that the adjacency matrix of constraints (1) is totally unimodular. An immediate implication of Lemma 1 is that we can solve this special case of the maintenance scheduling problem efficiently by solving the LP relaxation of the SCM formulation. The number of variables of SCM formulation can be reduced based on the following lemma.

**Lemma 2.** *There exists an optimal collection of disjoint spans in such a way that every span  $[\sigma_i, \sigma_k]$  in the collection has the following properties*

1.  $i = d'_j - p'_j + 1$  for some  $j \in \mathcal{J}$
2.  $k = d'_j - p'_j + p'_{j'}$  for some  $j, j' \in \mathcal{J}$  or  $k = r'_j + p'_j - 1$  for some  $j \in \mathcal{J}$

The above Lemma is proved by showing that from every optimal solution which does not have the stated properties, we can find another optimal solution which has the stated properties. As a consequence of the above Lemma, we can replace set  $F$  with subset

$$F' = \{(i, k) \in F : i = d'_j - p'_j + 1, k = d'_j - p'_j + p'_{j'} \text{ or } r'_j + p'_j - 1 \text{ for some } j, j' \in \mathcal{J}\}.$$

### 3 COMBINATORIAL ALGORITHMS

In this section, we present two combinatorial algorithms for solving the maintenance scheduling problem in the general case and in the special case that includes only up-paths. In the special case, we present a polynomial solution algorithm.

**Theorem 1.** *The maintenance scheduling problem is NP-complete*

We can prove NP-completeness by a reduction from minimum hitting of horizontal unit segments problem (Hassin and Megiddo (1991)). Since the maintenance scheduling problem is NP-complete, finding a polynomial time algorithm is impossible unless  $P = NP$ . In the following, we present a dynamic programming algorithm for the problem. Before proceeding, we need to define more notation. We define the following sets for  $P, X \subseteq \mathcal{P}$ :

$$\begin{aligned} R(X, P) &= \{\sigma_i^u \in P : \forall \sigma_j^u \in X, j < i\} \cup \{\sigma_i^d \in P : \forall \sigma_j^d \in X, j < i\}, \\ L(X, P) &= \{\sigma_i^u \in P : \forall \sigma_j^u \in X, j > i\} \cup \{\sigma_i^d \in P : \forall \sigma_j^d \in X, j > i\}, \\ J(P) &= \{j \in \mathcal{J} : \forall R \in \mathcal{R}_j, R \cap P \neq \emptyset\}, \\ J_M(X, P) &= \{j \in J(P) : \exists R \in \mathcal{R}_j, R \cap P \subseteq X \cap P\}, \\ J_R(X, P) &= \{j \in J(P) \setminus J_M(X, P) : \forall R \in \mathcal{R}_j, R \cap P \subseteq R(X, P) \cup (X \cap P)\}, \\ J_L(X, P) &= \{j \in J(P) \setminus J_M(X, P) : \forall R \in \mathcal{R}_j, R \cap P \subseteq L(X, P) \cup (X \cap P)\}. \end{aligned}$$

The set  $R(X, P)$  contains all paths in  $P$  that are on the right-hand side of paths in  $X$ . Similarly, the set  $L(X, P)$  contains all path in  $P$  that are on the left-hand side of paths in  $X$ . The set  $J(P)$  contains all jobs that will possess at least one path in  $P$  regardless of their start times.

**Definition 3.** Job  $j \in J(P)$  is called splitting if  $J(P) = J_L(R, P) \cup J_M(R, P) \cup J_R(R, P)$  for all  $R \in \mathcal{R}_j$ . Let  $J'(P)$  denote the set of splitting jobs in  $J(P)$ .

Let  $c(P)$  denote the minimum number of cancelled paths in the set  $P$  for scheduling jobs in  $J(P)$ . In particular,  $c(\mathcal{P})$  is the optimal objective value for the complete problem.

**Theorem 2.** *The optimal value  $c(\mathcal{P})$  (and a corresponding optimal solution) can be computed by the following recursion:*

$$c(P) = \begin{cases} 0 & \text{if } J(P) = \emptyset, \\ \min\{|X \cap P| + c(P \setminus X) : X \in \mathcal{R}_j\} & \text{for an arbitrary } j \in J(P) \\ \min\{|X \cap P| + c(R(X, P)) + c(L(X, P)) : X \in \mathcal{R}_j\} & \text{for an arbitrary } j \in J'(P) \end{cases}$$

If the traffic is only in one direction, the maintenance scheduling problem can be solved in polynomial time. We start by observing that we can restrict our attention to solutions satisfying a particular property.

**Lemma 3.** *There exists an optimal solution in which for each job  $j \in \mathcal{J}$ , the possession starts at  $\sigma_{r'_j}$  or  $\sigma_{d'_{j'}-p'_{j'}+1}$  for some  $j' \in \mathcal{J}$  and ends at  $\sigma_{r'_j+p'_j-1}$  or  $\sigma_{d'_{j'}-p'_{j'}+p_j}$  for some  $j' \in \mathcal{J}$ .*

The above Lemma is proved by using Lemma 2.

**Definition 4.** For job  $j \in \mathcal{J}$ , a path is called *essential* if it is equal to  $\sigma_{r'_j}$  or  $\sigma_{d'_{j'}-p'_{j'}+1}$  for some  $j' \in \mathcal{J}$ . For every  $j \in \mathcal{J}$ , set  $\mathcal{R}_j^e = \{[\sigma_a, \sigma_b] \in \mathcal{R}_j : \sigma_a \text{ is an essential path}\}$ .

The maintenance scheduling problem with unidirectional traffic and interval property is very close to the real time scheduling problem with infinite machine capacity in Khandekar et al. (2015). Their dynamic programming algorithm with small modifications can be applied in our setting and this leads to the following result. Let  $c(P)$  denote the minimum number of cancelled paths in the set  $P$  for scheduling jobs in  $J(P)$  in such a way that every job  $j \in J(P)$ . Following Lemma 3, it is clear that  $c(P)$  is equal to optimal value of maintenance scheduling problem with unidirectional traffic and interval property and a corresponding optimal solution, is an optimal solution for the problem.

**Theorem 3.** *For the maintenance scheduling problem with unidirectional traffic and interval property, the optimal value  $c(P)$  (and a corresponding optimal solution) can be computed in polynomial time using the following recursion. If  $J(P) = \emptyset$ , then  $c(P) = 0$ . Otherwise,*

$$c(P) = \min_{X \in \mathcal{R}_j^e} \{|X \cap P| + c(R(X, P)) + c(L(X, P))\},$$

where  $j \in J(P)$  is an arbitrary job of maximal length (i.e.,  $p'_j \geq p'_{j'}$  for all  $j' \in J(P)$ ).

#### 4 COMPUTATIONAL EXPERIMENTS

In this section, we compare MIP formulations on instances that include only up-paths and have the interval property. The computational experiments were carried out on a Dell Latitude E5570 laptop with Intel Core i7-6820HQ 2.70GHz processor, and 8GB of RAM running Ubuntu 16.04. We used Python 3.5 and Gurobi 7.0.2 to solve the MIP models. We set a time limit of 100 seconds for solving each instance. If an optimal solution was not reached within this time frame, the best incumbent was reported.

We compare the strength of each formulation on 12 classes of randomly generated instances. Each class contains 20 instances. An instance is given by a set of jobs  $\mathcal{J}$ , a set of paths  $\mathcal{P}$ , and a set of collections  $\mathcal{R}_j$ ,  $j \in \mathcal{J}$ . Each class is represented by a pair  $(n, L)$  where  $n$  is the number of jobs and  $L$  is the maximum number of distinct lengths. The main steps of the procedure for generating each instance of class  $(n, L)$  are as follows: First,  $L$  distinct lengths are chosen randomly from  $\{1, 2, 3, 5, 7, 11\}$ . Second, the number of paths (i.e.,  $m$ ) is set equal to  $n$  times the average of the chosen distinct lengths divided by 2. Third, the length of each job is chosen randomly from the  $L$  chosen distinct lengths. Finally, the first and last path for each job are randomly chosen from  $\{1, \dots, m\}$ .

We compare the models based on the following measures: the number of instances that are solved to optimality by the LP relaxation (N. LP Optimal), the average solution runtime (Avg. IP runtime), the maximum solution runtime (Max IP runtime), the average of

$$\frac{|\text{Optimal LP relaxation obj. value} - \text{Optimal obj. value}|}{\text{Optimal obj. value}}$$

(Avg. LP Gap), the maximum LP Gap (Max LP Gap), the average of

$$\frac{|\text{Incumbent obj. value} - \text{Optimal obj. value}|}{\text{Optimal obj. value}}$$

(Avg. MIP Gap) where incumbent refers to a best integer solution found within 100 seconds, and maximum MIP Gap (Max MIP Gap).

The results are presented in Tables 1 and 2. The LP relaxation of the HSM formulation did not yield the optimal value for any instance whereas the LP relaxation of SCM formulation gave the optimal value for the

**Table 1.** Computational results for SCM formulation. All times are wall clock times and are given in seconds.

$(n, Nl)$	N. LP Optimal	Avg. IP Runtime	Max IP Runtime	Avg. LP Gap (%)	Max LP Gap (%)
(30,2)	17	0.01	0.02	0.30	2.38
(30,3)	18	0.01	0.02	0.34	5.26
(30,4)	19	0.01	0.02	0.08	1.52
(50,2)	18	0.06	0.13	0.07	0.76
(50,3)	19	0.06	0.09	0.10	1.96
(50,4)	16	0.07	0.11	0.26	1.67
(70,2)	19	0.30	0.61	0.07	1.47
(70,3)	17	0.34	1.11	0.12	1.19
(70,4)	15	0.20	0.47	0.28	1.97
(100,2)	18	1.90	4.30	0.13	1.68
(100,3)	16	1.69	5.44	0.16	1.21
(100,4)	17	1.24	2.86	0.10	0.79

**Table 2.** Computational results for HSM formulation. All times are wall clock times and are given in seconds.

$(n, Nl)$	Avg. IP Runtime	Max IP Runtime	Avg. LP Gap (%)	Max LP Gap (%)	Avg. MIP Gap (%)	Max MIP Gap (%)
(30,2)	46.55	100.21	51.18	80.23	0	0
(30,3)	73.96	100.33	58.77	70.78	0.26	3.03
(30,4)	76.44	100.19	58.06	76.57	0.11	2.17
(50,2)	71.90	100.10	53.06	78.85	0.77	5.41
(50,3)	81.63	100.59	55.33	75.23	0.58	5.00
(50,4)	91.40	100.31	57.29	74.74	1.26	6.38
(70,2)	90.16	100.25	59.41	77.52	3.95	22.12
(70,3)	96.40	100.10	60.28	74.72	3.55	9.72
(70,4)	100.01	100.09	56.17	73.81	2.99	9.84
(100,2)	91.12	101.73	58.48	76.56	11.70	50.00
(100,3)	100.01	100.09	60.73	77.91	10.66	32.47
(100,4)	100.03	100.23	61.37	73.42	5.52	15.79

majority of instances. The maximum LP Gap for the SCM formulation on all instances is 5.26% whereas that of the HSM formulation is 80.23%. The maximum average LP Gap for the SCM formulation on all classes is 0.34% compared to 61.37% for the HSM formulation. The MIP gap on all instances for SCM formulation is zero whereas the maximum average and maximum MIP gap for HSM formulation on all classes are 11.70% and 50.00%. It is clear from the computational results that SCM formulation outperforms HSM formulation by high margin. We did some additional experiments for larger instances, and found that the SCM formulation typically finds the optimal solution within the time limit of 100 seconds for up to 380 jobs.

## 5 CONCLUSIONS

In this paper, we studied maintenance scheduling in a railway corridor. We showed that the problem is NP-complete. We presented an exact algorithm to solve the problem in Theorem 2 which has exponential time complexity in the worst case. In the special case when there is only unidirectional traffic, we presented a polynomial time algorithm. Furthermore, we presented a set covering formulation for this special case. The computational results shows its efficiency and strength.

## ACKNOWLEDGEMENT

This research was supported by the ARC Linkage Grant no. 140101000.

## REFERENCES

- Abed, F., L. Chen, Y. Disser, M. Groß, N. Megow, J. Meißner, A. T. Richter, and R. Rischke (2017). Scheduling maintenance jobs in networks. In *International Conference on Algorithms and Complexity*, pp. 19–30. Springer.
- BITRE (2016). *Australian Infrastructure Statistics Yearbook*. Bureau of Infrastructure, Transport and Regional Economics, Canberra ACT, Australia.
- Boland, N., T. Kalinowski, and S. Kaur (2015). Scheduling network maintenance jobs with release dates and deadlines to maximize total flow over time: Bounds and solution strategies. *Computers & Operations Research* 64, 113–129.
- Boland, N., T. Kalinowski, and S. Kaur (2016). Scheduling arc shut downs in a network to maximize flow over time with a bounded number of jobs per time period. *Journal of Combinatorial Optimization* 32(3), 885–905.
- Boland, N., T. Kalinowski, H. Waterer, and L. Zheng (2013). Mixed integer programming based maintenance scheduling for the Hunter Valley Coal Chain. *Journal of Scheduling* 16(6), 649–659.
- Boland, N., T. Kalinowski, H. Waterer, and L. Zheng (2014). Scheduling arc maintenance jobs in a network to maximize total flow over time. *Discrete Applied Mathematics* 163, 34–52.
- Boland, N., R. Kapoor, S. Kaur, and T. Kalinowski (2014). Scheduling unit time arc shutdowns to maximize network flow over time: complexity results. *Networks* 63(2), 196–202.
- Eskandarzadeh, S., T. Kalinowski, and H. Waterer (2017). Maintenance scheduling in a railway corridor. Working Paper.
- Hassin, R. and N. Megiddo (1991). Approximation algorithms for hitting objects with straight lines. *Discrete Applied Mathematics* 30(1), 29–42.
- Khandekar, R., B. Schieber, H. Shachnai, and T. Tamir (2015). Real-time scheduling to minimize machine busy times. *Journal of Scheduling* 18(6), 561.