# Maintenance location routing for rolling stock under line and fleet planning uncertainty

Denise D. Tönissen

School of Business and Economics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

d.d.tonissen@vu.nl

Joachim J. Arts

Luxembourg Centre for Logistics and Supply Chain Management, University of Luxembourg, Luxembourg City, Luxembourg; and School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands,

joachim.arts@uni.lu

Zuo-Jun (Max) Shen

Department of Industrial Engineering and Operations Research, Department of Civil and Environmental Engineering, Tsinghua-Berkeley Shenzhen Institute, University of California, Berkeley, Berkeley, California 94720, USA,

maxshen@berkeley.edu

August 4, 2018

**Abstract**

Rolling stock needs regular maintenance in a maintenance facility. Rolling stock from different fleets are routed to maintenance facilities by interchanging the destinations of trains at common stations and by using empty drives. We consider the problem of locating maintenance facilities in a railway network under uncertain or changing line planning, fleet planning and other uncertain factors. These uncertainties and changes are modeled by a discrete set of scenarios. We show that this new problem is NP-hard and provide a two-stage stochastic programming and a two-stage robust optimization formulation. The second-stage decision is a maintenance routing problem with similarity to a minimum cost-flow problem. We prove that the facility location decisions remain unchanged under a simplified routing problem and this gives rise to an efficient mixed integer programming (MIP) formulation. This result also allows us to find an efficient decomposition algorithm for the robust formulation based on scenario addition (SA). Computational work shows that our improved MIP formulation can efficiently solve instances of industrial size. SA improves the computational time for the robust formulation even further and can handle larger instances due to more efficient memory usage. Finally, we apply our algorithms on practical instances of the Netherlands Railways and give managerial insights.

**Keywords:** Two-stage optimization, column-and-constraint generation, facility location, rolling stock, maintenance routing

# 1 Introduction

We study the optimal location of maintenance facilities for rolling stock in a railway network. This new problem came to us through a 4 year collaboration with the Netherlands railways (NS). Like most facility location problems, there is a set of candidate facilities and their annual costs, and we have to decide which facilities to open. However, the *maintenance location routing problem* has essential features that make it substantially different from other facility location problems. In particular, the customers of facilities (train units) have to travel over a rigid railway network. The transportation costs of these train units are more intricate than can be modeled by fixed allocation costs. The problem of routing these train units to the maintenance facilities is called the maintenance routing problem (MRP). The MRP cannot be separated from the facility location problem because the ease with which a facility can be reached depends intricately on the railway infrastructure and the *line plan*.

A line plan consists of a set of train lines, where each line is a path in the railway network that is operated with a certain frequency by one rolling stock type. The line and fleet plan within a railway network change regularly to accommodate changing travel demands. As a consequence, any reasonable facility location plan must work well under a wide variety of line and fleet plan scenarios. This includes changes in how lines run, up and down-scaling of service frequencies on any given line, the rolling stock types assigned to the lines, and the introduction of new rolling stock types.

To deal with the features we outline above (maintenance location routing, line planning, and fleet planning), we provide two novel models to help managers decide where to locate maintenance facilities. These two models reflect different attitudes to risk that the decision maker may hold. The first model is a two-stage stochastic programming model that seeks to minimize the annual cost of the facilities and the annual maintenance routing cost averaged over a set of line plan scenarios. We call this the stochastic maintenance location routing problem (SMLRP). This model is appropriate for decision makers that are risk neutral. The second model is similar to the first, however it seeks to minimize the performance of the worst-case situation. Such an objective is often called a minimax or an absolute robust objective (Kouvelis and Yu, 2013). This model is therefore called the robust maintenance location routing problem (RMLRP) and can be seen as a two-stage robust optimization model (Ben-Tal et al., 2004). This model is appropriate for risk averse decision makers. The attitude of the NS towards risk is neither entirely risk neutral nor entirely risk averse. Therefore, they would like to study the solutions of both models.

The first-stage decision for the SMLRP and RMLRP is to open a facility, given a set of candidate facilities, annual facility depreciation costs, and a discrete set of line planning scenarios. In the second-stage the annual maintenance routing costs for a first-stage location decision are determined for each line plan scenario. The second-stage problem corresponds with the MRP. We give the deterministic equivalents of the two-stage models. The deterministic equivalents are multi-commodity flow models and we prove that exactly the same facilities are opened when we simplify these models to single-commodity flow models. This simplification decreases the number of variables and constraints for practical sized instances by millions. We call this formulation the improved mixed integer program (IMIP). The IMIP makes it possible to fit practical sized instances into memory and to solve them with CPLEX within reasonable time. Furthermore, we improve the computational time for the RMLRP further, by providing a scenario addition (SA) method. The SA method is a column-and-constraint (also called row-and-column) generation method that is specifically developed for problems with discrete scenario sets. SA adds the scenario constraints iteratively to the IMIP until an optimal solution is found, which improves the computational time and memory requirements for most instances. A similar idea is presented in Zeng and Zhao (2013) and Chan et al. (2017) for polyhedral uncertainty sets.

The main contributions of this paper are as follows.

1. We are the first to study a maintenance location problem on a railway network. In this setting it is necessary to incorporate the annual maintenance routing cost in the facility location decision. This leads to a new class of problems that we call maintenance location routing problems.

2. We provide a two-stage stochastic programming formulation and a two-stage robust optimization formulation for the maintenance location routing problem that can deal with uncertainties that are inherent in any application. In particular, our model can deal with uncertainties in line and fleet planning.

3. We provide a mixed integer formulation and an even more efficient SA algorithm that can solve instances of practical size.

4. We perform a case study for the Netherlands Railways (NS) and give managerial insights.

The paper starts with a literature review, followed by a detailed description of the MRP. In Section 4, we model the RMLRP and the SMLRP as two-stage problems and in Section 5.1 we provide our solution methodology. In Section 6, we perform computational experiments on randomly generated instances. We study the influence of the number of scenarios and facilities on the solution time and compare the algorithms. In Section 7, we present our case study for the NS and provide managerial insights.

## 2 Literature review

Facility location models have been studied extensively in the literature (Drezner and Hamacher, 2001; Laporte et al., 2016). The combination with operational supply chain decisions (Melo et al., 2009), vehicle routing (Nagy and Salhi, 2007), and uncertainty have also been studied in the literature (Snyder, 2006). However, the combination of facility location with maintenance routing has only been studied in the context of aviation applications (Feo and Bard, 1989; Gopalan, 2014). Furthermore, to the best of our knowledge, there are only four papers that combine maintenance with facility location (Lieckens et al., 2013; Rappold and Van Roo, 2009; van Ommeren and Bumb, 2006; Xie et al., 2016). However, the settings of these papers differ considerably from ours, and the authors use heuristics while we seek exact optimal solutions. We first describe the maintenance routing literature and we continue to the two-stage robust and stochastic facility location literature.

## 2.1 Maintenance routing

Many papers have been written about maintenance routing for railway applications. Anderegg et al. (2003), consider the situation in the German and Swiss Federal Railways where maintenance routing is part of the long-term vehicle scheduling problem. The authors use a minimum cost flow formulation that is often used for vehicle routing. Maintenance cannot be adapted into the flow model and a heuristic modification is used to satisfy the maintenance constraints. Maróti and Kroon (2005, 2007) consider maintenance routing for the NS, where maintenance routing is not part of the vehicle scheduling problem. They use a two- to five-day time window during which a train unit is routed to the maintenance facility. The main reason for this is that timetables and rolling stock schedules are dense, and that there are many disturbances. Consequently, long-term models cannot take shunting issues and disturbances into account. The authors formulate this NP-hard problem as two different multicommodity flow problems, and they solve the models with CPLEX. Other papers consider variants of the locomotive planning problem. The locomotive planning problem assigns locomotives to a set of train units in such a way that it minimizes the cost and satisfies a number of business and operational constraints. For a recent survey on the locomotive assignment problem, that also includes variants with some maintenance constraints, see Piu and Speranza (2014). Furthermore, there are many papers about maintenance routing for aviation such as those of Clarke et al. (1997), Talluri (1998), and Sarac et al. (2006).

## 2.2 Two-stage robust and stochastic facility location

Uncertainty for facility location models can be classified into three categories (Shen et al., 2011): receiver-side uncertainty; in-between uncertainty; and provider-side uncertainty. Like most stochastic facility location models (see the references in Snyder (2006) and Swamy and Shmoys (2006)), our paper focuses on the first two uncertainties. These uncertainties are related to customer uncertainty (for example, customer demand or customer location) and incomplete knowledge about the transportation network topology, transportation times, or costs between facilities and customers. A common feature of the receiver-side and in-between uncertainties is that the uncertainty does not change the topology of the provider-receiver network once the facilities have been built. Our problem does not share this feature as the different line plans do change the topology of the provider-receiver network; a different line plan changes the network over which rolling stock are routed to a maintenance facility.

The SMLRP bears some resemblance to supply chain network design under uncertainty, which includes the location of facilities within a supply chain. Santoso et al. (2005) design a supply chain network consisting of suppliers, processing facilities, and customers under cost, demand, supply and capacity uncertainty. A sample average approximation is used to generate a discrete set of scenarios and the model is solved with Benders decomposition. Santoso et al. (2005) describe and test many acceleration techniques, that we also implement in the online appendix of this paper. The accelerated Benders decomposition method works well for their problem, but is less successful for our model as compared with the IMIP. Other examples of Benders decomposition for supply chain network design include fixed charge network design (Costa, 2005), freight-forwarding network design (Üster and Agrahari, 2011), reverse supply chain design (Santibanez-Gonzalez and Diabat, 2013), and closed loop supply chain network design (Khatami et al., 2015).

Two types of methods are generally used for two-stage robust problems. The first method is similar to Benders decomposition and uses constraint generation based on the dual information of the slave problem. Álvarez-Miranda et al. (2015), use Benders decomposition accelerated with additional cuts and a primal heuristic on a two-stage robust facility location problem with a discrete set of scenarios. An important difference to our paper is that their allocation of customers to facilities does not include the routing of customers (rolling stock) to (maintenance) facilities. Another difference is the considered uncertainty and recoverability of the model. In our model, we open facilities in the first-stage and allocate customers to these facilities in the second-stage. Álvarez-Miranda et al. (2015) open facilities and assign customers in the first-stage and the second-stage is used to recover the facilities and customer assignment to the revealed scenario. Gabrel et al. (2014) solve a non-linear convex two-stage robust location transportation problem. In this problem a commodity has to be transported from each of the $m$ potential sources to each of the $n$ destinations. The demand of the destinations is uncertain. The authors use a cutting plane algorithm based on Benders decomposition, where the slave problem is NP-hard. Furthermore, Benders decomposition has been applied to two-stage robust unit commitment problems (Bertsimas et al., 2013; Jiang et al., 2011).

The second method consists of column-and-constraint generation procedures. Zeng and Zhao (2013) show that a column-and-constraint generation procedure solves a two-stage robust location transportation problem with demand levels in a polyhedral uncertainty set an order of magnitude faster than Benders decomposition. Chan et al. (2017) apply a column-and-constraint generation procedure to a robust facility location problem where the demand points are uncertain and An et al. (2014) apply it to the reliable p-median facility location problem. The strategy is also used for two-stage robust unit commitment (An and Zeng, 2015; Zhao and Zeng, 2012) and a two-stage robust distribution network reconfiguration problem (Lee et al., 2015).

# 3 Maintenance routing

The goal of the MRP is to determine the annual routing costs for train units to enter given maintenance facilities with given capacities. A *maintenance facility* is a facility that is responsible for the planned inspections and maintenance of rolling stock. The frequency of inspections and maintenance is dependent on the rolling stock type and typically occurs once every half year up to every month. The maintenance routing is influenced by the current *line plan*, a set of routes (paths) in a rail network, operated with a certain frequency by a specific rolling stock type. The stations where a line starts or ends are called *end stations*. At an end station, all passengers leave the train and the train unit drives back or continues on another line after a break. Between the end stations, a line often has many other regular train stations where passengers can leave or enter the train. The transport from the train lines to the maintenance facilities is done with help of *interchanges*. An interchange swaps the destinations of two train units of the same rolling stock type that are on connecting train lines. Connecting train lines share an end station and at that end station the train unit can be interchanged. The train units continue on each other's train line after such an interchange. A train that requires maintenance is interchanged with another train unit until it reaches a train line connected to a maintenance facility. When a maintenance facility cannot be reached by a specific train unit via these interchanges, *deadheading* is used for the remaining trip. Deadheading is driving with an empty train (no passengers), and is undesirable. Deadheading gives additional driving cost and the train unit is not available for public transport, which can result in shorter trains and passenger discomfort.

The operational problem of routing specific train units to their maintenance facilities is studied by Maróti and Kroon (2005, 2007). The operational maintenance routing problem is solved daily and its decisions are based on operational information such as the arrival and departure time of train units and the shunting possibilities of the end stations. In this section, we introduce a new maintenance routing model that determines the annual operational maintenance routing cost by taking maintenance routing decisions on an aggregate level. This enables us to combine the maintenance routing model with facility location decisions in Section 4. Our objective is to minimize the annual interchange and deadheading cost. Consequently, we do not consider separate train units, but work with the average number of maintenance visits departing from each line per year, the routes for each visit, and the annual number of interchanges.

## 3.1 Problem description

Given is a physical rail network $G_P = (N_P, E_P)$, consisting of nodes (stations) $N_P$ and edges (railway tracks) $E_P$, a set of end stations $E \subseteq N_P$, and a set with opened facilities $O \subseteq E$, where every opened facility has a capacity that represents the maximum number of maintenance visits the location can handle per year. Furthermore, we are given a line plan that consists of a set of lines $L$, with for each line the type of rolling stock, the maintenance frequency per year, its end stations and the deadheading cost to each maintenance facility. The line plan also specifies the set of possible interchanges, with a coordination cost for each interchange, end station interchange capacities, and a network interchange budget.

The network interchange budget $G$ restricts the annual number of interchanges in the network. The interchange budget is highly dependent on the frequently changing rolling stock schedule and it is possible to improve certain interchanges by making small changes to the rolling stock schedule. Modeling this interchange budget allows us to gain insight into which interchanges should be made possible or improved, while also constraining the number of interchanges that can be used. Furthermore, we constrain the annual number of interchanges at each end station.

In the left-hand side of Figure 1, we show an example of a physical railway network. Here we exclude all stations that are not an end station in at least one of the line plans shown in the middle and on the right of Figure 1. For the MRP, only one line plan for the network $G_P$ is given. However, in Section 4, the different line plans for network $G_P$ play an important role. Each line (edge) connects two end stations (nodes). The number of end stations can differ between the line plans; e.g. station $W$ is an end station in the middle of Figure 1, while it is an in-between station of the line $(X, V)$ on the right-hand side. Furthermore, there are two rolling stock types in this example. Rolling stock type $a$, is a regional train, stopping at every small station, while type $b$ is an intercity train that only stops at the large cities. An example of an interchange for the middle picture is line $(U, V)$ to line $(V, W)$, while an interchange from $(U, V)$ to line $(U, W)$ is not possible because the rolling stock types do not match.

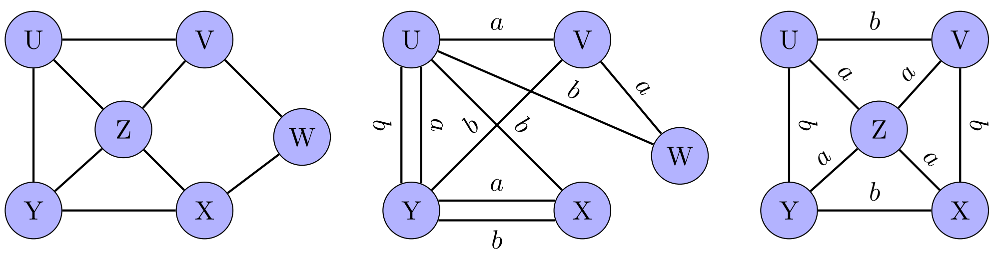



Figure 1: The physical rail network on the left and two line plan possibilities.

## 3.2 Maintenance routing model

We model the MRP as a flow-based model. We only allow interchanges followed by deadheading directly to the already opened maintenance facilities. The reason for this is practical, and not a restriction of the model. Routes with deadheading followed by interchanges are not often used in practice, can be very expensive, and cause imbalances in the number of train units per line, which need to be solved. Consequently, we do not include these kinds of routes in our model. We use the line plan and physical railway network to create a directed maintenance routing flow graph $G_F = (N_F, A_F)$.

- For every line, we make a node; the set with these nodes is denoted by $N_L \subset N_F$.
- We create one source $\mathcal{S}$ that is connected with a directed arc to each node in $N_L$.
- We create an arc between lines where an interchange is possible. The cost of this arc are the interchange coordination cost. The set of these interchange arcs is denoted by $A_I \subset A_F$.
- We create a node for every open facility and denote the set of open facilities by $N_O \subset N_F$. Each node in $N_O$ is connected with an arc to the sink $\mathcal{T}$.

- For each node $n \in N_L$, we make an arc to each facility. The cost of the arc is 0 when a line is connected to the facility and it is the deadheading costs from a line to a facility when it is not connected. The set of these incoming facility arcs is denoted by $A_O \subset A_F$.

In Figure 2, we demonstrate how to apply this transformation. The left-hand side depicts a line plan for the physical railway network in Figure 1, where $a$ and $b$ are the rolling stock types and 0-8 are the line numbers. The right-hand side depicts the maintenance routing flow graph $G_F = (N_F, A_F)$ for the line plan on the left when facilities $N_O = \{U, W, Y\}$ are opened.

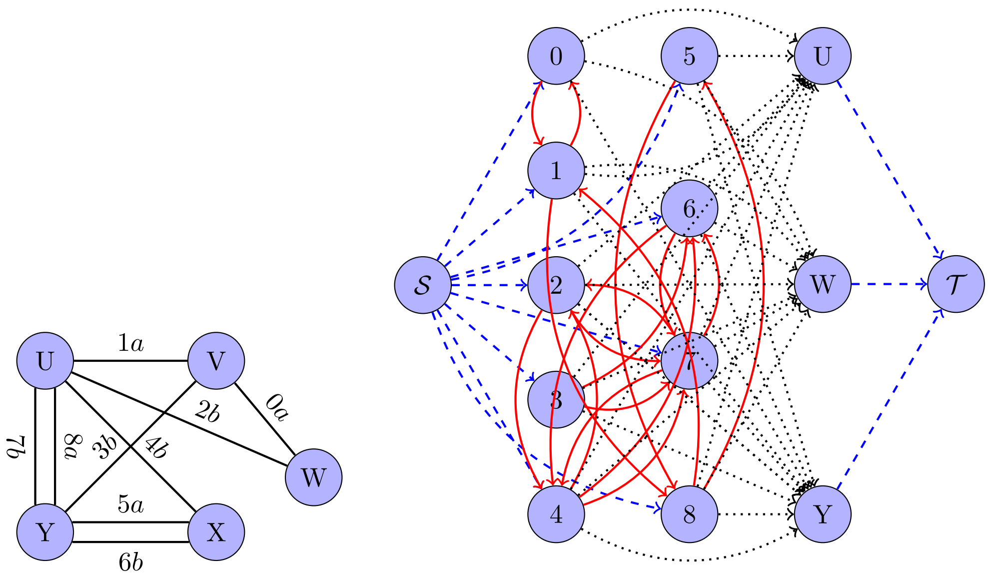



Figure 2: Left a possible line plan for the physical railway network in Figure 1. Right the resulting maintenance routing flow graph ($G_F = (N_F, A_F)$). The arcs from and to the source $\mathcal{S}$ and sink $\mathcal{T}$ are dashed blue, the interchange arcs ($A_I$) solid red, and the arcs to the facilities ($A_O$) are dotted black.

The aforementioned additional combinations of interchanges and deadheading can easily be allowed by creating an additional arc from every line node to every other line node, which represents the deadheading from one line to another.

**Proposition 1.** *The number of nodes and arcs in the maintenance routing flow graph is polynomial in the number of lines and end stations.*

*Proof.* The number of nodes in the maintenance routing flow graph is equal to $|L| + |N_O| + 2$. The number of arcs is equal to $|L| + |L||N_O| + |A_I| + |N_O|$, where $|A_I|$ is bounded by $|L|(|L| - 1)$. □

The *maintenance frequency* of a line $l$ is the annual number of train units originating from that line that require maintenance. The maintenance frequency for line $l \in L$ is defined by the parameter $m_l$, and $n_l \in N_L$ is the node associated with line $l$. The flow through arc $a \in A_F$ that is associated with the yearly maintenance frequency originating from line $l \in L$ is represented by the decision variable $z_l(a)$. For example, $z_4(2, 7)$ represents the frequency of interchanges from line 2 to line 7 for a maintenance visit originating from line 4, and $z_1(8, y)$ represents the frequency of maintenance visits originating from line 1 that reach maintenance facility $y$ via line 8.

We define $\delta_{\text{in}}(n)$ and $\delta_{\text{out}}(n)$ as the set of ingoing and outgoing arcs of node $n \in N_F$, and the set of arcs representing the interchanges going through end station $e$ is defined as $A_e \subset A_I$. The cost of arc $a$ for flow of type $l$ is $c_l(a)$, and is only defined for the arcs that represent the interchanges and the deadheading $(A_I \cup A_O)$. The capacity for the facilities, $q_n$ $\forall n \in N_O$, is measured by the yearly maintenance frequency that the facility can handle. Furthermore, the number of interchanges at end station $e \in E$ is restricted by the parameter $g_e$, and the annual number of maintenance visits over the entire network by $G$.

We summarize the notation in Table 1 and formulate the following linear program:

$$(\text{MRP}) \quad \min \sum_{a \in A_I \cup A_O} \sum_{l \in L} c_l(a) z_l(a)$$

$$\text{s.t.} \sum_{a \in \delta_{\text{in}}(n)} \sum_{l \in L} z_l(a) \leq q_n \qquad \forall n \in N_O, \tag{1}$$

$$\sum_{a \in \delta_{\text{in}}(n)} z_l(a) = \sum_{a \in \delta_{\text{out}}(n)} z_l(a) \qquad \forall n \in N_F \setminus \{\mathcal{S}, \mathcal{T}\}, \quad \forall l \in L, \tag{2}$$

$$z_l(a) = m_l \qquad \forall l \in L, \quad a \in \delta_{\text{in}}(n_l) \setminus A_I, \tag{3}$$

$$\sum_{a \in \delta_{\text{in}}(\mathcal{T})} \sum_{l \in L} z_l(a) = \sum_{l \in L} m_l, \tag{4}$$

$$\sum_{a \in A_e} \sum_{l \in L} z_l(a) \leq g_e \qquad \forall e \in E, \tag{5}$$

$$\sum_{a \in A_I} \sum_{l \in L} z_l(a) \leq G, \tag{6}$$

$$z_l(a) \geq 0 \qquad \forall a \in A_F, \quad \forall l \in L. \tag{7}$$

Constraints (1) restrict the number of yearly maintenance visits that can be assigned to opened facility $n$ to its capacity. Constraints (2) are the flow conservation constraints, while Constraints (3) and (4) guarantee that every maintenance visit is assigned to a facility. Constraints (4) are necessary to exclude alternative solutions with more flow than the number of yearly maintenance visits; these solutions are possible because some of the routes to the facilities have zero costs. Constraints (5) and (6) are the end station and interchange budget constraints.

We can solve the MRP in polynomial time because it is a linear program. Furthermore, the problem is similar to the minimum cost flow problem with the exception of the multiple commodities $l \in L$ and Constraints (1), (5), and (6). We show that even with these additional constraints the MRP has an optimal integer solution $z_l(a)$ if $g_e$ $(\forall e \in E)$, $G$, $m_l$ $(\forall l \in L)$, and $q_n$ $(\forall n \in N_F)$ are integers.

**Proposition 2.** *If $g_e$ $(\forall e \in E)$, $G$, $m_l$ $(\forall l \in L)$, and $q_n$ $(\forall n \in N_F)$ are integers, then for any basic feasible solution of the MRP each flow $z_l(a)$ $(\forall a \in A_F, \forall l \in L)$ is an integer.*

The proof of Proposition 2 can be found in online Appendix A.

| | |
|---|---|
| Graph notation and sets: | |
| $G_P = (N_P, E_P)$ | physical railway network consisting of nodes $N_P$ and edges $E_P$ |
| $E$ | set with end stations ($E \subseteq N_P$) |
| $O$ | set with opened locations ($O \subseteq E$) |
| $G_F = (N_F, A_F)$ | maintenance routing flow graph consisting of nodes $N_F$ and arcs $A_F$ |
| $\mathcal{S}$ | source node |
| $\mathcal{T}$ | sink node |
| $N_O$ | set of nodes associated with the opened facilities $O$ ($N_O \subset N_F$) |
| $L$ | set with all train lines |
| $N_L$ | set of nodes associated with the lines $L$ ($N_L \subset N_F$) |
| $n_l$ | the node associated with line $l$ |
| $A_O$ | set of arcs that go to the opened facility nodes ($A_O \subset A_F$) |
| $A_I$ | set of interchange arcs ($A_I \subset A_F$) |
| $A_e$ | set of interchange arcs for end station $e$ ($A_e \subset A_I$) |
| $\delta_{\text{in}}(n)$ | set containing the incoming arcs of node $n$ ($\delta_{\text{in}}(n) \subset A_F$) |
| $\delta_{\text{out}}(n)$ | set containing the outcoming arcs of node $n$ ($\delta_{\text{out}}(n) \subset A_F$) |
| Parameters: | |
| $m_l$ | maintenance frequency for line $l \in L$ |
| $c_l(a)$ | cost for using arc $a$ for a maintenance visit originating from line $l \in L$ |
| $q_n$ | capacity of facility $n \in N_O$ |
| $g_e$ | interchange budget of end station $e \in E$ |
| $G$ | network interchange budget |
| Decision variables: | |
| $z_l(a)$ | maintenance frequency decision variable for arc $a$ and line $l$ |

Table 1: Graph notation, sets, parameters, and decision variables for the MRP.

# 4 Maintenance location routing

In this section, we introduce the SMRLP and RMRLP. We model both problems as two-stage models. We use a discrete scenario set $D$, in which every scenario defines a line plan (see also Figure 1). The first-stage decision is the decision to open a facility, given a set of candidate facilities, annual facility depreciation costs, and a discrete set of scenarios. The second-stage decision is taken for each scenario and corresponds to the MRP discussed in Section 3. The RMLRP optimizes the cost for the worst-case scenario $d \in D$, and the SMLRP optimizes the cost for the average case.

The SMLRP and RMLRP are NP-hard because the capacitated facility location problem (CFLP) is a special case of these problems. In the CFLP, a set of demand points, a set of facilities, the capacities of the locations and their cost, and the cost of assigning a demand point to a facility are given. The objective is to open a set of facilities and to assign each demand point to a facility, while not exceeding the capacity of each facility and minimizing the cost. When our model has only one scenario and the network interchange budget for this scenario is 0, then only deadheading from the lines to the locations is possible. When we interpret the lines as demand points, with the required maintenance visit frequency as demand, then the deadheading cost is exactly the same as the assignment costs, and the SMLRP/RMLRP can be used to solve the CFLP.

## 4.1 The two-stage models

As in Section 3, we are given a physical rail network $G_P = (N_P, E_P)$, consisting of nodes (stations) $N_P$ and edges (railway tracks) $E_P$. Furthermore, we are given a set of discrete scenarios $d \in D$, in which each scenario defines a line plan: a set of lines $L^d$ $\forall d \in D$, with for each line the type of rolling stock and the maintenance frequency per year, the end stations $E^d \subseteq N_P$ $\forall d \in D$, and their location in the physical rail network. A line plan for a scenario specifies the set of possible interchanges, with a coordination cost for each interchange, the deadheading cost from each line to each facility, the end station interchange capacities, and the interchange network budget. Finally, we are given a set of candidate facilities $C$, with for every candidate facility a capacity $q_n$ $\forall n \in N_C$, that represents the maximum annual maintenance frequency the location can handle. The first-stage decision is represented by the binary decision variable $y \in \{0,1\}^{|C|}$, which is 1 when a facility is opened and 0 otherwise.

For every first-stage decision $y \in \{0,1\}^{|C|}$ and scenario $d \in D$, there is a directed maintenance location routing flow graph $G_F^{dy} = (N_F^{dy}, A_F^{dy})$. The notation of the variables and parameters is similar to those in Section 3. The maintenance frequency for line $\forall l \in L$ and scenario $d \in D$ is defined by the parameter $m_l^d$, and $n_l^d$ is the node associated with line $l$ and scenario $d$. The flow through arc $a$ associated with the annual maintenance frequency from line $l \in L^d$, in scenario $d \in D$, is represented by the decision variable $z_l^d(a)$. We define $\delta_{\text{in}}^d(n)$ and $\delta_{\text{out}}^d(n)$ as the set of ingoing and outgoing arcs of node $n$ for scenario $d$. Here $A_I^d$ is the set of interchange arcs and $A_C^d = \bigcup_{n \in N_C} \delta_{\text{in}}^d(n)$, the set of arcs going to the candidate facilities. The cost of arc $a$ for line $l$ and scenario $d$ is $c_l^d(a)$. The arc costs are only defined for arcs in the set $\bigcup_{d \in D}(A_I^d \cup A_C^d)$. The end station interchange budget $g_e^d$ is a capacity on the interchange frequency at end station $e \in E^d$ for scenario $d \in D$. The set of arcs representing the interchanges going through end station $e$ for scenario $d$ is denoted by $A_e^d$. Furthermore, the number of annual interchanges in the entire network for each scenario $d \in D$ is restricted by the parameter $G^d$. The weights $w_d$ $\forall d \in D$ denote the expected fraction of time that a line plan is used during the life time of the facilities and is only relevant for the SMLRP. We formulate the following two-stage models:

$$(\text{SMLRP-2SSP}) \quad \min \sum_{n \in N_C} c_n y_n + \sum_{d \in D} w_d \mathrm{MRP}_d(y)$$

$$(\text{RMLRP-2SRO}) \quad \min \sum_{n \in N_C} c_n y_n + \max_{d \in D} \mathrm{MRP}_d(y)$$

$$\text{s.t.} \sum_{n \in N_F} y_n q_n \geq \max_{d \in D} \sum_{l \in L^d} m_l^d, \tag{8}$$

$$y_n \in \{0,1\} \qquad \forall n \in N_C, \tag{9}$$

where

$$\mathrm{MRP}_d(y) = \min \sum_{a \in A_I^d \cup A_C^d} \sum_{l \in L^d} c_l^d(a) z_l^d(a)$$

$$\text{s.t.} \sum_{a \in \delta_{\text{in}}^d(n)} \sum_{l \in L^d} z_l^d(a) \leq y_n q_n \qquad \forall n \in N_C, \tag{10}$$

$$\sum_{a \in \delta_{\text{in}}^d(n)} z_l^d(a) = \sum_{a \in \delta_{\text{out}}^d(n)} z_l^d(a) \qquad \forall n \in N_F^{dy} \setminus \{\mathcal{S}, \mathcal{T}\}, \quad \forall l \in L^d, \tag{11}$$

$$z_l^d(a) = m_l^d \qquad \forall l \in L^d, \quad a \in \delta_{\text{in}}^d(n_l^d) \setminus A_I, \tag{12}$$

$$\sum_{a \in \delta_{\text{in}}^d(\mathcal{T})} \sum_{l \in L^d} z_l^d(a) = \sum_{l \in L} m_l^d, \tag{13}$$

$$\sum_{a \in A_e^d} \sum_{l \in L^d} z_l^d(a) \leq g_e^d \qquad \forall e \in E^d, \tag{14}$$

$$\sum_{a \in A_I^d} \sum_{l \in L^d} z_l^d(a) \leq G^d, \tag{15}$$

$$z_l^d(a) \geq 0 \qquad \forall a \in A_F^{dy}, \quad \forall l \in L^d. \tag{16}$$

The objectives of the SMLRP-2SSP and RMLRP-2SRO formulations optimize the annual facility cost in combination with the average/worst-case maintenance routing cost, under Constraint (8). Constraint (8) guarantees that the opened facilities have sufficient combined capacity to handle all maintenance visits, so that the second-stage problem has a feasible solution. Here $\mathrm{MRP}_d(y)$ optimizes the maintenance routing cost for scenario $d$ and first-stage decision $y$. Constraints (10) guarantee that maintenance visits can only be assigned to opened facilities up to their capacity. Constraints (11) are the flow conservation constraints, while Constraints (12) and (13) guarantee that every maintenance visit is assigned to a facility. Constraints (14) and (15) are the end station and network interchange budgets constraints.

The recourse matrices of the SMLRP-2SSP and RMLRP-2SRO do not have a fixed size which is different from most two-stage problems. The network topology for each $\mathrm{MRP}_d(y)$ is different and as a consequence the values and dimensions of the recourse matrices are different for each scenario. Furthermore, Constraint (8) guarantees that each feasible first-stage solution is feasible for the $\mathrm{MRP}_d(y)$ for any scenario $d \in D$. Consequently, we have relatively complete recourse.

# 5 Solution methodology

The two-stage formulations of Section 4 are not practical in terms of computation. In this section, we reformulate these problems into a mixed integer programming formulation, that is also known as the deterministic equivalent or extensive form (Birge and Louveaux, 1997). Then we provide an improved mixed integer programming formulation (IMIP) that is equivalent in terms of the optimal objective and optimal facility decisions, but not with respect to the second-stage decisions. This improved formulation is very efficient in finding the optimal facility location decisions.

In addition, we design a column-and-constraint generation algorithm (similar to Zeng and Zhao, 2013; Chan et al., 2017) for the RMLRP. Our method, called scenario addition (SA), solves the IMIP with a subset of scenarios and adds an additional scenario in each iteration. The algorithm reaches optimality in a finite number of iterations. SA uses the fact that for the RMLRP only the facility solution $y^*$ and the worst-case scenario are required for the solution.

We also applied accelerated Benders decomposition to the SMLRP and RMLRP. However, accelerated Benders decomposition performed very poorly in the computational experiments and therefore the details of this approach are reported in online Appendix B.

## 5.1 Mixed integer programming formulation

The first step to reformulate the two-stage models SMLRP-2SSP and RMLRP-2SRO to mixed integer programming formulations is to make one large directed maintenance location routing flow graph. This graph contains all scenarios, instead of one graph for every scenario. The steps used in Section 3 are adapted to generate the directed graph $G_M = (N_M, A_M)$ as follows.

- For every scenario and line, we make a node. The set with all line nodes belonging to a scenario $d \in D$ is denoted by $N_L^d$.
- We make one source $\mathcal{S}$ that is shared for all scenarios. The source is connected with an arc to each node in $\bigcup_{d \in D} N_L^d$.
- We make an arc between every line where an interchange is possible (connected, same rolling stock type, same scenario), and the cost is the interchange coordination cost.
- We make a node for every candidate facility, and each of these nodes is connected with an arc to the sink $\mathcal{T}$. The set of candidate facility nodes is denoted by $N_C$.
- For each node $n \in \bigcup_{d \in D} N_L^d$, we make an arc to each facility. The cost of this arc is the deadheading cost of the line to the facility. The cost of the arc is zero when deadheading is not necessary because the line associated with the node is connected to the facility.

The sets $N_M^d$ and $A_M^d$ contain all nodes or arcs that can be reached by flow from scenario $d \in D$. Furthermore, by Proposition 2, the flow $z_l^d(a)$ will be integer when $g_e$ ($\forall e \in E$), $G$, $m_l$ ($\forall l \in L$), and $q_n$

($\forall n \in N_F$) are integers. The two-stage models SMLRP-2SSP and RMLRP-2SRO can be reformulated to the following mixed integer programming formulations:

$$\text{(SMLRP-MIP)} \quad \min \sum_{n \in N_C} c_n y_n + \sum_{d \in D} w_d \sum_{a \in A_I^d \cup A_C^d} \sum_{l \in L^d} c_l^d(a) z_l^d(a)$$

$$\text{(RMLRP-MIP)} \quad \min \sum_{n \in N_C} c_n y_n + z_{\max}$$

$$\text{s.t.} \sum_{a \in \delta_{\text{in}}^d(n)} \sum_{l \in L^d} z_l^d(a) \leq y_n q_n \qquad \forall d \in D, \quad \forall n \in N_C, \tag{17}$$

$$\sum_{a \in \delta_{\text{in}}^d(n)} z_l^d(a) = \sum_{a \in \delta_{\text{out}}(n)} z_l^d(a) \qquad \forall d \in D, \quad \forall n \in N_M^d \setminus \{\mathcal{S}, \mathcal{T}\}, \quad \forall l \in L^d, \tag{18}$$

$$z_l^d(a) = m_l^d \qquad \forall d \in D, \quad \forall l \in L^d, \quad \forall a \in \delta_{\text{in}}^d(n_l^d) \setminus A_I, \tag{19}$$

$$\sum_{d \in D} \sum_{a \in \delta_{\text{in}}^d(\mathcal{T})} \sum_{l \in L^d} z_l^d(a) = \sum_{d \in D} \sum_{l \in L^d} m_l^d, \tag{20}$$

$$\sum_{a \in A_e^d} \sum_{l \in L^d} z_l^d(a) \leq g_e^d \qquad \forall d \in D, \quad \forall e \in E^d, \tag{21}$$

$$\sum_{a \in A_I^d} \sum_{l \in L^d} z_l^d(a) \leq G^d \qquad \forall d \in D, \tag{22}$$

$$z_l^d(a) \geq 0 \qquad \forall d \in D, \quad \forall a \in A_M^d, \quad \forall l \in L^d, \tag{23}$$

$$y_n \in \{0, 1\} \qquad \forall n \in N_C, \tag{24}$$

with the following additional constraints for the RMLRP-MIP:

$$z_{\max} \geq \sum_{a \in A_I^d \cup A_C^d} \sum_{l \in L^d} c_l^d(a) z_l^d(a) \qquad \forall d \in D. \tag{25}$$

Constraints (17)-(23) are comparable to (10)-(16) from Section 4, however now defined for the graph $G_M = (N_M, A_M)$ and for all scenarios. Note that in the maintenance location routing flow graph $G_M$, all arcs besides those going to the sink can only be traversed by flow from one scenario. Consequently, by the use of $A_M^d$ and $N_M^d$ in place of $A_M$ and $N_M$, we have removed many unnecessary constraints in Constraints (18) and (23). Constraints (25), which are only used for the RMLRP-MIP, guarantee that $z_{\max}$ equals the routing cost of the worst-case scenario. The SMLRP-2SSP and SMLRP-MIP are equivalent. However, for the RMLRP-2SRO and RMLRP-MIP only the opened and closed facilities $y^*$ and the objective value are equivalent. Because we are not interested in the second-stage decisions, the RMLRP-MIP can be used as if it is equivalent to the RMLRP-2SRO.

## 5.2 Improved mixed integer programming formulation

The RMLRP-MIP and SMLRP-MIP require so much available memory that only very small instances can be solved. However, the required memory can be significantly reduced by two observations. The first observation is that we do not need to distinguish the scenario for each flow. Without the index $d$, we can still determine to which scenario a flow belongs based on the incoming arcs of the facility nodes as they always belong to only one scenario. For the second observation, we use practical knowledge of the cost structure of the problem. For practical instances the cost of an interchange or deadheading ($c_l^d(a)\ \forall a \in A_I^d \cup A_C^d$) depends on the rolling stock type and the scenario, and not on the originating line. As a consequence $c_m^d(a) = c_n^d(a)\ \forall m, n \in L$, when the rolling stock type assigned to line $m$ and $n$ are the same. Because it is not allowed to interchange train units with different rolling stock types, dropping the $l$ index will not change the cost. Consequently, the incoming flow per facility will be the same, resulting in the same facility decisions and the same optimal objective value.

Furthermore, Constraints (20) can be omitted, because without the $l$ indices, all flow is already restricted by Constraints (19). Additionally, in Constraints (17) $q_n$ can be much higher than the total number of maintenance visits ($\sum_{l \in L^d} m_l^d$) for a scenario $d \in D$. Tightening Constraints (17), by replacing $q_n$ by $\hat{q_n^d} = \min(q_n, \sum_{l \in L^d} m_l^d)$ for every scenario $d$ improves the initial LP bound. Finally, we denote $\bigcup_{d \in D} \delta_{\text{in}}^d(n)$ and $\bigcup_{d \in D} \delta_{\text{out}}^d(n)$ as $\delta_{\text{in}}(n)$ and $\delta_{\text{out}}(n)$ respectively. This gives the following improved mixed integer programming (IMIP) formulations:

$$\text{(SMLRP-IMIP)} \quad \min \sum_{n \in N_C} c_n y_n + \sum_{d \in D} w_d \sum_{a \in A_I^d \cup A_C^d} c(a) z(a)$$

$$\text{(RMLRP-IMIP)} \quad \min \sum_{n \in N_C} c_n y_n + z_{\max}$$

$$\text{s.t.} \sum_{a \in \delta_{\text{in}}^d(n)} z(a) \le y_n \hat{q_n^d} \qquad \forall d \in D, \quad \forall n \in N_C, \tag{26}$$

$$\sum_{a \in \delta_{\text{in}}(n)} z(a) = \sum_{a \in \delta_{\text{out}}(n)} z(a) \qquad \forall n \in N_M \setminus \{\mathcal{S}, \mathcal{T}\}, \tag{27}$$

$$z(a) = m_l^d \qquad \forall d \in D, \quad \forall l \in L^d, \quad \forall a \in \delta_{\text{in}}^d(n_l^d) \setminus A_I, \tag{28}$$

$$\sum_{a \in A_e^d} z(a) \le g_e^d \qquad \forall d \in D, \quad \forall e \in E^d, \tag{29}$$

$$\sum_{a \in A_I^d} z(a) \le G^d \qquad \forall d \in D, \tag{30}$$

$$z(a) \ge 0 \qquad \forall a \in A_M, \tag{31}$$

$$y_n \in \{0, 1\} \qquad \forall n \in N_C, \tag{32}$$

with the following additional constraints for the (RMLRP-IMIP):

$$z_{\max} \ge \sum_{a \in A_I^d \cup A_C^d} c(a) z(a) \qquad \forall d \in D. \tag{33}$$

Note that the $l$ index is still required for Constraints (28). The reason for this is that although during routing we need not be concerned with the origin of the flow, we still need to make sure that the right amount of flow starts from each origin. As a consequence, the parameter $m_l^d$ is needed to specify the amount of flow that originates from each line.

**Theorem 1.** *The SMLRP-IMIP and RMLRP-IMIP have identical optimal costs and facility decisions $y^*$ as the SMLRP-MIP and RMLRP-MIP, respectively, when $c^d_m(a) = c^d_n(a)$ $\forall m, n \in L$ if the rolling stock types assigned to line $m$ and $n$ are the same.*

*Proof.* For every $a \in \delta_{\text{out}}(\mathcal{S})$, $z^d_l(a) \geq 0$ for only one scenario $d$ and line $l$, and for all other scenarios and lines $z^d_l(a) = 0$. This is the case because of Constraints (19) and (20) that guarantee that every line node receives only $m^d_l$ flow for one $l$ and $d$ and 0 for all other lines and scenarios. These line nodes are only connected to the line nodes of the same scenario and the facility nodes. Consequently, the line nodes can only receive flow of one scenario type, and the scenario of the incoming flow of the facilities, is equal to the scenario to which the line node from that the incoming facility flow originates from belongs to. Furthermore, dropping the indices will not influence the amount of flow and cost for every arc. The amount of flow for each arc $z(a)$ will become equal to $\sum_{d \in D} \sum_{l \in L^d} z^d_l(a)$. Furthermore, the cost of the flow remains the same because every arc $a \in A^d_I \cup A^d_C$ can only be reached by maintenance visits of exactly one rolling stock type and scenario. Consequently, the same facilities will be opened and the objective value will be the same. □

**Theorem 2.** *The SMLRP-IMIP and RMLRP-IMIP reduce the number of variables by $\Theta(\sum_{d \in D} \left(|L^d|^2|N_C| + |L^d||A^d_I|\right))$ compared with the SMLRP-MIP and RMLRP-MIP, respectively.*

*Proof.* Note that $|A_M| = \sum_{d \in D}(|L^d| + |L^d||N_C| + |A^d_I|) + |N_C|$ and $|A^d_M| = |L^d| + |L^d||N_C| + |A^d_I| + |N_C|$.

We subtract the number of variables in the IMIP from the number of MIP variables. The number of removed variables is

$$
\begin{aligned}
&\sum_{d \in D} \left(|L^d||A^d_M|\right) - |A_M| \\
&= \sum_{d \in D} \left(|L^d|(|L^d| + |L^d||N_C| + |A^d_I| + |N_C|)\right) - \sum_{d \in D}(|L^d| + |L^d||N_C| + |A^d_I|) - |N_C| \\
&= \sum_{d \in D} \left(|L^d|^2 + |L^d|^2|N_C| + |L^d||A^d_I| + |L^d||N_C|\right) - \sum_{d \in D}(|L^d| + |L^d||N_C| + |A^d_I|) - |N_C| \\
&= \sum_{d \in D} \left(|L^d|^2 + |L^d|^2|N_C| + |L^d||A^d_I| + |L^d||N_C| - |L^d| - |L^d||N_C| - |A^d_I|\right) - |N_C| \\
&= \sum_{d \in D} \left(|L^d|^2 + |L^d|^2|N_C| + |L^d||A^d_I| - |L^d| - |A^d_I|\right) - |N_C|
\end{aligned}
$$

Consequently, the number of removed variables is of the order of $\Theta(\sum_{d \in D} \left(|L^d|^2|N_C| + |L^d||A^d_I|\right))$. □

**Theorem 3.** *The SMLRP-IMIP and RMLRP-IMIP reduce the number of constraints by $\Theta(\sum_{d \in D} \left(|L^d|^2 + |N_C||L^d|\right))$ compared with the SMLRP-MIP and RMLRP-MIP, respectively.*

*Proof.* We subtract the number of constraints in the IMIP from the number of MIP constraints. Note that we removed Constraint (20) and that only Constraints (18) and (27) are different. Furthermore, $|N_M| = \sum_{d \in D} |L^d| + |N_C| + 2$ and $|N^d_M| = |L^d| + |N_C| + 2$. The number of removed constraints is

$$\sum_{d \in D} \left(|L^d|(|N^d_M| - 2)\right) + 1 - |N_M| - 2$$

$$= \sum_{d \in D} \left(|L^d|^2 + |N_C||L^d|\right) + 1 - \left(\sum_{d \in D} |L^d| + |N_C|\right)$$
$$= \sum_{d \in D} \left(|L^d|^2 + |N_C||L^d| - |L^d|\right) - |N_C| + 1.$$

Consequently, the number of removed constraints is of order $\Theta(\sum_{d \in D} \left(|L^d|^2 + |N_C||L^d|\right))$. □

To summarize, the IMIP formulation reduces the number of variables and constraints substantially and simplifies a multi-commodity model to a single-commodity model. As a consequence, the required memory and solution time are significantly reduced.

## 5.3 Scenario addition

It is necessary to solve the second-stage problem (MRP) from Section 4 many times for SA. Therefore, we use the improvements from Section 5.2 for the MRP and introduce the IMRP. Although the MRP can be solved in polynomial time, this change yields a substantial reduction in solution time. For larger instances (100 facilities, 64 scenarios), the solution time decreases from multiple seconds per scenario, to solving all scenarios in approximately one second. Furthermore, similar to Section 5.1, the capacity $q_n$, can be replaced by $\hat{q}_n^d$.

These changes together give us the following improved maintenance routing problem formulation:

$$\text{IMRP}_d(y) = \min \sum_{a \in A_I^d \cup A_C^d} c(a)z(a)$$

$$\begin{aligned}
\text{s.t.} \sum_{a \in \delta_{\text{in}}(n)} z(a) &\leq y_n \hat{q}_n^d && \forall n \in N_C, && (\mu)\\
\sum_{a \in \delta_{\text{in}}(n)} z(a) &= \sum_{a \in \delta_{\text{out}}(n)} z(a) && \forall n \in N_F^{dy} \setminus \{\mathcal{S}, \mathcal{T}\}, && (\nu)\\
z(a) &= m_l^d && \forall l \in L^d, \quad a \in \delta_{\text{in}}^d(n_l^d) \setminus A_I, && (\pi)\\
\sum_{a \in A_e^d} z(a) &\leq g_e^d && \forall e \in E^d, && (\phi)\\
\sum_{a \in A_I^d} z(a) &\leq G^d, && && (\omega)\\
z(a) &\geq 0 && \forall a \in A_F^{dy}.
\end{aligned}$$

The Greek symbols next to the constraints are the corresponding dual variables, which are used for the Benders decomposition algorithm described in online Appendix B.

SA uses the fact that for the RMLRP, only the opened facilities and the worst-case scenario are relevant for the solution. Consequently, SA cannot be used for the SMLRP where all scenarios contribute to the objective function. Let $i$ be an iteration counter that starts at 0 and $D^i$ denote the scenario set used in iteration $i$. The set $D^0$ contains one randomly chosen scenario $d \in D$. Because the scenarios differ in the total number of maintenance visits, the following feasibility constraint is added to the RMLRP-IMIP:

$$\sum_{n \in N_C} y_n \hat{q}_n \geq \max_{d \in D} \sum_{l \in L^d} m_l^d, \tag{34}$$

where $\hat{q}_n = \min\{q_n, \max_{d \in D} \sum_{l \in L^d} m_l^d\}$.

With these preliminaries, the SA algorithm consists of the following steps.

1. Compute the solution to the RMLRP-IMIP with $D$ replaced by $D^i$, and add Constraint (34) to the formulation. Denote the optimal objective of this problem as $\mathrm{LB}^i$ and the solution of iteration $i$ as $y^i$.

2. Set $\mathrm{UB}^i := c^T y^i + \max_{d \in D} \mathrm{IMRP}_d(y^i)$ and let $d^i_{\max} := \underset{d \in D}{\arg\max}\ \mathrm{IMRP}_d(y^i)$. If $d^i_{\max} \in D^i$, then $\mathrm{UB}^i = \mathrm{LB}^i$. Stop and return $y^i$ as the optimal solution and $\mathrm{UB}^i$ as the optimal objective value. Otherwise ($d^i_{\max} \notin D^i$) the algorithm proceeds to the next step.

3. Let $D^{i+1} := \{d^i_{\max}\} \cup D^i$. Update $i := i + 1$ and go back to Step 1.

These steps give a very diverse set of scenarios as the most detrimental solution for the incumbent solution is added in each iteration. We observe that the algorithm generally converges quickly with only a small number of scenarios in $D^i$.

# 6 Computational experiments

In this section, we report computational experiments on randomly generated instances to test the computational performance of our algorithms. We are particularly interested in the size of instances that can be solved by the IMIP and SA. The results of our accelerated Benders decomposition algorithm can be found in the online Appendix B. Although we generate instances randomly, the fixed and random parameters are based on those found in practice to create reasonable instances. All experiments are programmed in Java with the CPLEX library version 12.6.3, and run on a laptop with an Intel Core i7-4710MQ Quad Core 2.5 GHz processor with 8 GB of RAM. All mentioned solution times include the time necessary to build the model and CPLEX standard settings are used (e.g. for preprocessing, branching, accuracy tolerance etc.).

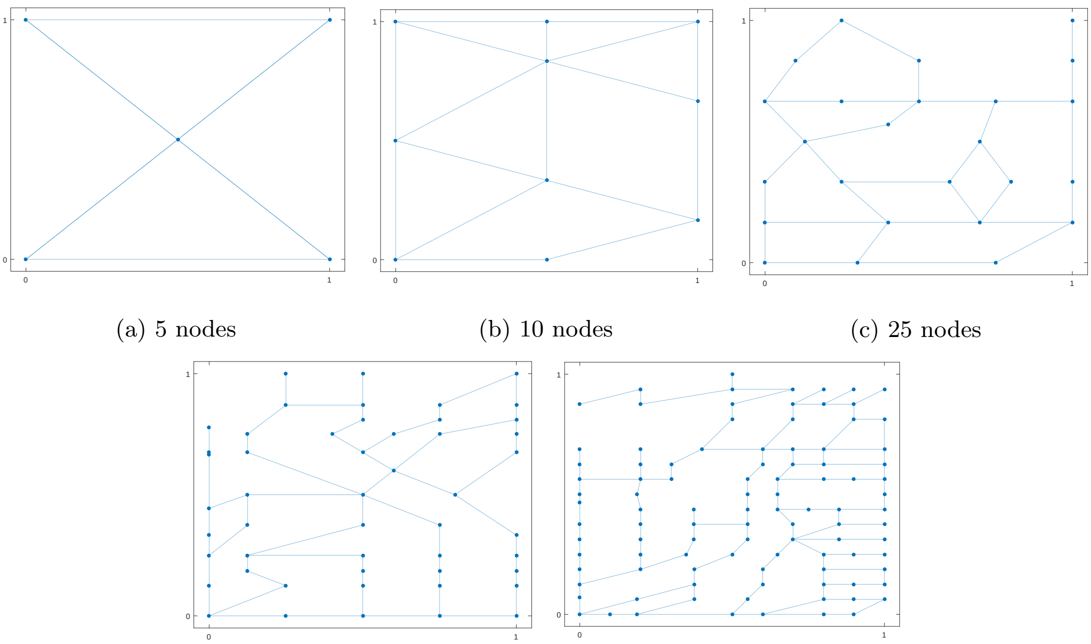
## 6.1 Test instance generation

An instance of the SMLRP or the RMLRP requires a railway network, several line plans on this railway network, parameters such as the sizes of facilities, interchange budgets etc. Subsection 6.1.1 explains how we create physical railway networks and possible assignments of end stations within the line plan (basic line plan). Then in Section 6.1.2 we explain how we generate many line planning scenarios for a physical railway network with basic line plans. Finally, Section 6.1.3 explains how interchange budgets and candidate locations for an instance are generated.

### 6.1.1 Physical railway network and basic line plan generation

We create random instances by first generating graphs of physical railway networks in the Cartesian plane. We do this by manually generating five archetypical graphs on a one by one plane with 5, 10, 25, 50, and 100 nodes, respectively, as shown in Figure 3. Each node is a potential end station and candidate facility location, while each edge is a set of rails between end stations. For each archetypical railway network, we also manually design "basic line plan" scenarios. A basic line plan consists only of origin-destination pairs and the route traveled between them. We also determine for each basic line whether the corresponding rolling stock type will be regional or intercity. Note, however, that there may be several regional and intercity rolling stock types, and that the actual rolling stock type will be assigned in Section 6.1.2.

We create physical railway networks by perturbing and scaling these archetypical networks. We perturb these archetypical graphs by relocating each node uniformly at random within a square around that node. The square is centered around the original node except at the boundaries of the $1 \times 1$ plane where the square is relocated to fit within the $1 \times 1$ plane. The dimension of this square decreases with the number of nodes in the network. Next, the obtained graph is scaled to either a (1) $200 \times 400$ km, (2) $750 \times 1000$ km, or (3) $3000 \times 3000$ km rectangle to model different railway network sizes. The length of railway tracks (edges) is then determined by multiplying the Euclidean length by a number drawn uniformly between 1.0 and 1.2. Note that at this point, the length in kilometers of each basic line is fixed. A graph thus obtained is denoted $G_P = (N_P, E_P)$.



(a) 5 nodes

(b) 10 nodes

(c) 25 nodes

(d) 50 nodes

(e) 100 nodes

Figure 3: The five archetypical graphs.

### 6.1.2 Line plan generation

Recall that a train line consists of a rolling stock type, the maintenance frequency per year and the deadheading cost to each facility. Before we can generate a line plan, we first need to generate different types of rolling stock. We first generate the number of different regional and intercity rolling stock types. This number is generated uniformly between 1 and the number of edges $|N_P|$ divided by 8. The deadheading cost per kilometer for each rolling stock type is calculated by generating the components of which it consists: the fuel, driver salary, unavailability costs, and fees. Each component is generated uniformly at random between the lower and upper bounds that are seen in practice for regional and intercity trains. Furthermore, the calculation of the number of maintenance visits for a line later on requires the annual number of maintenance visits of the rolling stock type on that line. The annual number of maintenance visits is generated uniformly at random between 1 and 12 for each rolling stock type; which are bounds that we observed at the NS.

Next, we generate line plan scenarios until we have reached the fixed number of scenarios for the instance. To generate a line plan scenario, we first take a random line plan possibility from the fixed basic line plan set. For each line of the chosen basic line plan, we generate the rolling stock type and maintenance visit frequency. The line is either an intercity or regional train line, and the rolling stock type is generated from the corresponding set. The rolling stock type is the same as that of a connecting line of the same line type with a probability of 0.5 for each line. When there are no defined lines yet, or when the rolling stock type will not be the same as an already defined connecting line, we choose uniformly at random from the set with rolling stock types for the basic rolling stock type of the line.

The annual maintenance frequency is the average number of train units on that line times the yearly maintenance frequency of the rolling stock type used for that line. An estimation for the average number of train units for a train line is made based on its length in kilometers, the average speed of the trains and the hourly train frequency of the line, which is generated uniformly at random between 1 and 8 (the lowest and highest hourly train frequencies considered by the NS). The deadheading cost for a line to a facility is the deadheading cost per kilometer of the rolling stock type times the number of kilometers for the shortest path in $G_P$ from one of the end stations nodes to the facility node. For example, when one of the end stations is the candidate facility, the shortest path is 0 kilometers and the cost is 0.

### 6.1.3 Interchange budgets and candidate facilities

All interchange coordination costs between lines are set at 10 euros. We define $\mathcal{U}(a, b)$ as a uniform random variable on $(a, b)$. Furthermore, we define the total annual number of maintenance visits for end station $e$ as $C_e^d := \sum_{l \in L_e^d} m_l^d$, where $L_e^d$ contains all lines that have $e$ as an end station.

The interchange budget for end station $e$ in scenario $d$ is randomly generated as

$$
g_e^d \coloneqq \begin{cases} 0 & \text{with a probability of } 0.05, \\ \mathcal{U}(0,1)C_e^d & \text{with a probability of } 0.2, \\ \mathcal{U}(1,3)C_e^d & \text{with a probability of } 0.65, \\ \infty & \text{with a probability of } 0.1. \end{cases}
$$

Note that $\mathcal{U}(1,3)C_e^d$ is only a restriction when there are many interchanges originating from non-connecting lines. We define $M^d \coloneqq \sum_{l \in L^d} m_l^d$ and randomly generate $G^d$ as

$$
G^d \coloneqq \begin{cases} \mathcal{U}(0,1)M^d & \text{with a probability of } 0.25, \\ \mathcal{U}(1,3)M^d & \text{with a probability of } 0.65, \\ \infty & \text{with a probability of } 0.1. \end{cases}
$$

The annual facility costs are generated uniformly at random between 1/5th and 5 times the estimated average annual facility cost for the NS. We define $M^{\text{dmax}} \coloneqq \max_{d \in D} \sum_{l \in L^d} m_l^d$ and generate the maintenance facility capacities as

$$
q_n \coloneqq \begin{cases} \mathcal{U}(0,0.5)M^{\text{dmax}} & \text{with a probability of } 0.1, \\ \mathcal{U}(0.5,1)M^{\text{dmax}} & \text{with a probability of } 0.4, \\ \infty & \text{with a probability of } 0.5. \end{cases}
$$

Consequently, on average half of the maintenance facilities have no capacity restriction. For the SMLRP, we also generate the weights $w_d$. These weights are generated uniformly at random between 0 and 1, followed by scaling the weights such that they sum up to 1.

## 6.2 Computational results for the IMIP

The test instance generation method from Section 6.1 can generate instances with any number of candidate facilities and line planning scenarios. For our computational tests, we created a full factorial of instances where the number of candidate facilities takes values in $\{5, 10, 25, 50, 100\}$ and the number of line planning scenarios takes values of $2^i$ for $i \in \{0, 1, \ldots, 15\}$. For each combination of number of facilities and line planning scenarios, 15 instances are generated. These 15 instances are split equally between the 3 archetypical railway networks in Section 6.1.1.

For each number of candidate facilities, we try to solve each instance within an hour. When at least 80% (13 out of 15) of the instances can be solved, we move to the set of instances where the number of scenarios is doubled. When an instance cannot be solved within an hour, a fail is registered and a solution time of 3600 seconds is used for the average time calculations. The average time in seconds for each set for the SMLRP and RMLRP can be found in Table 2. The top row shows the number of candidate facilities per set and the first column the number of scenarios.

| | SMLRP-IMIP | | | | | RMLRP-IMIP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Scenarios** | **5** | **10** | **25** | **50** | **100** | **5** | **10** | **25** | **50** | **100** |
| 1 | $< 0.1$ | $< 0.1$ | 0.1 | 1.1 | 19.4 | $< 0.1$ | $< 0.1$ | 0.1 | 0.7 | 13.4 |
| 2 | $< 0.1$ | $< 0.1$ | 0.2 | 1.7 | 110 | $< 0.1$ | $< 0.1$ | 0.2 | 1.8 | 103.4 |
| 4 | $< 0.1$ | $< 0.1$ | 0.3 | 8.6 | 94.2 | $< 0.1$ | 0.1 | 0.6 | 9.1 | 175.5 |
| 8 | $< 0.1$ | 0.1 | 0.6 | 20.6 | 202.6 | 0.1 | 0.1 | 2.2 | 35.2 | 874.3 |
| 16 | 0.1 | 0.2 | 2.5 | 33.2 | 956.6 | 0.2 | 0.2 | 2.8 | 67.6 | fails |
| 32 | 0.1 | 0.3 | 5.7 | 112.5 | fails | 0.2 | 0.3 | 5.3 | 231.0 | |
| 64 | 0.2 | 0.5 | 10.4 | 296.7 | | 0.3 | 0.6 | 13.5 | 715.4 | |
| 128 | 0.4 | 1.4 | 37.6 | 785.8 | | 0.5 | 1.7 | 71.4 | 1259.3 | |
| 256 | 1.1 | 3.0 | 112.9 | fails | | 1.6 | 4.2 | 140.1 | fails | |
| 512 | 2.9 | 11.8 | 507.4 | | | 5.0 | 15.2 | 452 | | |
| 1024 | 10.1 | 36.0 | 1076.8 | | | 16.3 | 44.9 | 1259.8 | | |
| 2048 | 31.1 | 154.8 | fails | | | 54.7 | 178.5 | fails | | |
| 4096 | 126.8 | 653.4 | | | | 214.4 | 531.2 | | | |
| 8192 | 556.7 | fails | | | | 733.2 | 1978.1 | | | |
| 16384 | 2428.3 | | | | | fails | fails | | | |
| 32768 | fails | | | | | | | | | |

Table 2: Solution time in seconds for the SMLRP-IMIP and RMLRP-IMIP, while varying the number of scenarios and facilities.

For up to 50 facilities, the IMIP solved with CPLEX works well. Note that these instances are already large. An instance with 25 facilities and 1024 scenarios has on average 35k (35765.5) nodes and a million (1007822.6) edges in the maintenance location routing flow graph for the RMLRP. Furthermore, the CPLEX model has a million (1007848.6) columns and 100k (100123.1) rows on average. For 100 candidate facilities, we can only solve the instances with a limited number of scenarios within an hour. For these instances, we extended the solution time to 48 hours (2880 minutes), to explore the boundary of what is computationally feasible. Because of time considerations, we limit the instances of the sets to the first six instances, two instances from each size. These results are shown in Table 3, where RAM is used for instance sets where an out-of-memory error occurred.

Instances with up to 64 scenarios can generally be solved. Furthermore, the SMLRP-IMIP seems to be computationally faster than the RMLRP-IMIP. A possible explanation for this are Constraints (33) from Section 5.2, which are only used for the RMLRP-IMIP. These constraints create degeneracy, because with them only one scenario contributes to the transportation cost, giving many solutions with the same objective value.

Furthermore, we note that the IMIP (solved with CPLEX standard branch-and-cut algorithm) uses

| | SMLRP-IMIP | | | | RMLRP-IMIP | | | |
|---|---|---|---|---|---|---|---|---|
| **Scenarios** | **Min** | **Average** | **Max** | **Failed** | **Min** | **Average** | **Max** | **Failed** |
| 16 | 0.4 | 12.1 | 56.2 | 0 | 17.4 | 105.6 | 252.6 | 0 |
| 32 | 1.5 | 72.3 | 331.5 | 0 | 22.8 | 429.6 | 791.4 | 0 |
| 64 | 4.6 | 209.2 | 892.7 | 0 | 123.0 | 631.8 | 2541.0 | 0 |
| 128 | 6.6 | 595.9 | > 2880.0 | 2 | 168.7 | 2337.7 | > 2880.0 | 4 |
| 256 | 18.5 | 774.5 | > 2880.0 | 1 | RAM | RAM | RAM | 6 |
| 512 | RAM | RAM | RAM | 6 | RAM | RAM | RAM | 6 |

Table 3: Results for instances with 100 facilities.

many cuts and few nodes in the branch and bound tree. Some of the instances can be solved without branching. For example, the average number of nodes for the SMLRP with 50 candidate facilities is 72, while it uses 4226 cuts on average. The number of nodes increases with the number of candidate facilities, while the number of cuts increases by both the number of facilities and scenarios. CPLEX mainly uses flow and MIP rounding cuts, followed by implied bound cuts, and to a lesser extent clique, Gomory, zero-half, and lift and project cuts.

## 6.3 Computational results for SA

We compare SA with the IMIP for the instances with 25, 50, and 100 candidate facilities from the randomly generated instance set. The instances with 5 and 10 facilities are excluded because the IMIP can already solve instances with up to 8192 scenarios within an hour. Again more than 80% (13 or more out of 15) of the instances have to be solved within an hour to go to the next set. The average time in seconds for each set for the RMLRP-IMIP and SA can be found in Table 4; the top row shows the number of candidate facilities per set and the first column the number of scenarios.

For the instances with 25 and 50 candidate facilities, it can be seen in Table 4 that the SA algorithm can solve instances with more scenarios within the hour. For the instances with 25 candidate facilities, we stopped the test after 32768 scenarios, although they can still easily be solved within the hour. There is no difference between SA and IMIP for the instances with 100 facilities. This can be explained by the fact that we can only solve instances with a limited number of scenarios for instances with 100 facilities. Consequently, an SA iteration with fewer scenarios added already takes a considerable amount of time.

When we extend the time from 1 hour to 48 hours (we limit the sets to the first six instances, as explained in Section 6.2), the SA algorithm performs better in both the best and average case, but its worst-case performance is as expected worse. We show these results in Table 5.

The number of iterations (is equal to the number of added scenarios) that the SA algorithm uses and the time in seconds are shown in Table 6. The number of iterations initially grows when the number of scenarios increases, but stabilizes at some point. The maximum number of iterations is low compared

| | IMIP | | | SA | | |
|---|---|---|---|---|---|---|
| **Scenarios** | **25** | **50** | **100** | **25** | **50** | **100** |
| 1 | 0.1 | 0.7 | 13.4 | 0.1 | 0.7 | 13.5 |
| 2 | 0.2 | 1.8 | 103.4 | 0.3 | 2.4 | 129.1 |
| 4 | 0.6 | 9.1 | 175.5 | 1.2 | 14.1 | 220.9 |
| 8 | 2.2 | 35.2 | 874.3 | 2.7 | 42.2 | 719.7 |
| 16 | 2.8 | 67.6 | fails | 4.4 | 76.4 | fails |
| 32 | 5.3 | 231.0 | | 5.1 | 328.7 | |
| 64 | 13.5 | 715.4 | | 5.3 | 357.4 | |
| 128 | 71.4 | 1259.3 | | 8.4 | 267.9 | |
| 256 | 140.1 | fails | | 11.9 | 485.3 | |
| 512 | 452 | | | 12.8 | 528.9 | |
| 1024 | 1259.8 | | | 19.1 | 851.9 | |
| 2048 | fails | | | 40.5 | 503.6 | |
| 4096 | | | | 54.9 | 1095.0 | |
| 8192 | | | | 75.3 | 1073.2 | |
| 16384 | | | | 144.3 | fails | |
| 32768 | | | | 288.8 | | |

Table 4: Solution time in seconds for the IMIP and SA, while varying the number of scenarios and facilities.

| | RMLRP-IMIP | | | | SA | | | |
|---|---|---|---|---|---|---|---|---|
| **Scenarios** | **Min** | **Average** | **Max** | **Failed** | **Min** | **Average** | **Max** | **Failed** |
| 16 | 17.4 | 105.6 | 252.6 | 0 | 1.2 | 74.4 | 198.0 | 0 |
| 32 | 22.8 | 429.6 | 791.4 | 0 | 5.4 | 560.4 | 2247.0 | 0 |
| 64 | 123.0 | 631.8 | 2541.0 | 0 | 8.4 | 511.2 | $> 2880.0$ | 1 |
| 128 | 168.7 | 2337.7 | $> 2880.0$ | 4 | 2.4 | 807.0 | $> 2880.0$ | 2 |
| 256 | RAM | RAM | RAM | 6 | 66.0 | 1103.7 | $> 2880.0$ | 2 |

Table 5: Results for instances with 100 facilities.

with the total number of scenarios $|D|$ of the instances. The maximum number of scenarios for 25 facilities is 32768, while for the worst-case, the scenario set only contains 20 scenarios. When we increase the number of candidate facilities to 50, the maximum number of scenarios is 8192, while the maximum number of iterations is 24.

| | 25 | | 50 | | 100 | |
|---|---|---|---|---|---|---|
| **Scenarios** | **Iterations** | **Time** | **Iterations** | **Time** | **Iterations** | **Time** |
| 1 | 1.0 | 0.1 | 1.0 | 0.7 | 1 | 13.5 |
| 2 | 1.9 | 0.3 | 1.9 | 2.4 | 2 | 129.1 |
| 4 | 2.9 | 1.2 | 3.5 | 14.1 | 3.4 | 220.9 |
| 8 | 4.6 | 2.7 | 5.3 | 42.2 | 4.9 | 719.7 |
| 16 | 5.9 | 4.4 | 6.9 | 76.4 | | |
| 32 | 7.3 | 5.1 | 9.3 | 328.7 | | |
| 64 | 7.8 | 5.3 | 11.7 | 357.4 | | |
| 128 | 7.1 | 8.4 | 11.2 | 267.9 | | |
| 256 | 7.5 | 11.9 | 11.8 | 485.3 | | |
| 512 | 7.9 | 12.8 | 10.9 | 528.9 | | |
| 1024 | 7.9 | 19.1 | 10.9 | 851.9 | | |
| 2048 | 8.1 | 40.5 | 11.6 | 503.6 | | |
| 4096 | 8.3 | 54.9 | 12.5 | 1095.0 | | |
| 8192 | 7.0 | 75.3 | 12.1 | 1073.2 | | |
| 16384 | 6.7 | 144.3 | | | | |
| 32768 | 6.7 | 288.8 | | | | |
| **max** | **21** | **983.1** | **24** | **> 3600** | **8** | **> 3600** |

Table 6: The average and maximum number of iterations needed for SA.

# 7 Case study: NS

In this section, we perform a case study for the NS. We use a green field approach, where existing facilities of the NS are kept out of scope. We first describe the NS instances and continue with our experiments. In our experiments, we investigate the benefit of including interchanges, the expected value of perfect information, the value of the stochastic or robust solution, and provide a robust solution for the case study.

## 7.1 NS instances

The NS instances each have 59 end stations and 55 candidate facilities. Four end stations (Utrecht, Breukelen, Amsterdam, and Schiphol) have been excluded, because a maintenance facility cannot be built at these locations. The facility costs are an estimation of the average annual cost of land, the necessary infrastructure, and the maintenance facility itself including all side buildings. Furthermore, we altered the facility cost based on which province the end station is located. The cost is either decreased or increased dependent on the province's average land price. The capacity restrictions of the facilities unfortunately only become clear in a later stage when a chosen location is investigated in detail. Because also no rough estimations could be given by the experts of the NS, we assume unlimited capacities for the facilities such that managerial insights can still be provided. (We conduct a sensitivity analysis on this assumption later and find that facility capacities are not an important issue.) Furthermore, we assume unlimited interchange capacities for all end stations except Utrecht, Amsterdam, and Schiphol, where interchanges are not possible. The network interchange budget has a different value for each experiment, and will be described in detail for each experiment.

We use four basic line plans: the current situation (2015), an estimation of 2018, and two possibilities for approximately 2025. These basic line plans contain all the lines (97, 97, 99, and 100 lines) the rolling stock serving the line and an estimate of the number of annual maintenance visits per line. The future scenarios, are based on the plan "Beter en Meer" (Prorail and NS, 2014), where the NS intends to increase the frequency of the lines in the *Randstad* or *larger Randstad*. The Randstad is a megalopolis in the Netherlands consisting of the four largest cities and their surrounding areas. The larger Randstad also includes the cities Arnhem, 's-Hertogenbosch, and Eindhoven.

The rolling stock types consist of all current rolling stock types and the future rolling stock types (FLIRT, SNG, and ICNG). For each rolling stock type, a rough estimation is made for the deadheading cost per kilometer. This estimation is based on many components such as driver cost, energy consumption, an estimation on the average deadheading velocity, and availability costs. The availability costs are based on the life cycle cost of the rolling stock and the passenger dissatisfaction cost due to the unavailability of the train. The cost of the interchanges is set at 10 euro per interchange.

Scenarios are made by picking a basic line plan, and altering the number of maintenance visits and rolling stock type. For each line of the line plan, the number of maintenance visits is from a triangular distribution, where the number of maintenance visits of the basic line plan is the mode of this distribution. We assume that the number of maintenance visits can decrease with 32.5% and increase with 75%, due to uncertainty in the annual number of maintenance visits of the train unit and the number of passengers using a certain line. A maximum of 20% of the rolling stock types of the lines can be swapped with each other. To do that, we generate for both regional and intercity trains a separated integer list, with its rolling stock number in the list for every maintenance visit it makes. These lists and the lines are shuffled. Then we go through all train lines and with a 10% probability, we change the rolling stock type to the first different rolling stock type of the list, all previous items of the list are removed. The maintenance frequency of the line is divided by the maintenance frequency of the current rolling stock type and multiplied by those of the new rolling stock type. We stop when there are no more lines or the limit of 20% has been reached.

The information that is available about the expected duration for each scenario is too limited to make any reasonable estimation. As a consequence, to demonstrate the SMLRP and to be able to compare a weighted solution (SMLRP) with a worst-case solution (RMLRP), we generate the weights randomly. We generate the weights $w_d$ uniformly at random between 0 and 1, followed by scaling the weights such that they sum up to 1.

## 7.2 Experiments

In this subsection, we compare the SMLRP and RMLRP and we provide managerial insights. We test the influence of including maintenance routing by varying the network interchange budget. We continue with the expected value of perfect information and the stochastic/robust solution and we provide a solution to the case study. In addition, we test the influence of the number of scenarios in online Appendix C.

Before we can test the influence of the network interchange budget, we need to know how many scenarios those instances should have. Consequently, we start with some initial computational experiments. We do those experiments with sets of 10 instances, increasing the number of scenarios by a factor 2 for each set. We fixed the network interchange budget for every scenario at $0.75M$, where $M = \sum_{l \in L^{2015}} m_l^{2015}$, which is the number of maintenance visits in the current (2015) scenario. For the RMLRP instances, the most often opened facilities are Eindhoven and Almere Oostvaarders (less than 32 scenarios), Eindhoven and Hilversum (64 up to 4096 scenarios) and 's-Hertogenbosch and Hoofddorp for 8192 or more scenarios. However, for each number of scenarios, there are always other solutions. We evaluated the mentioned three solutions, with 10 instances that have 65536 scenarios, the difference between the optimal solution values is at most 2%. This small difference explains why we obtain different solutions, even if we include many scenarios. We show these solutions in Figure 4, within the first black circle are the stations in the north Randstad (Hilversum, Hoofddorp, and Almere Oostvaarders) and the stations in the south of the larger Randstad (Eindhoven, 's-Hertogenbosch) are in the second black circle. From now on, we use instances with 8192 scenarios that can be solved in approximately 10 minutes.

When we relax the infinite capacity assumption and vary the capacities of the facilities, the solution remains the same when the capacity of the facilities are larger than $1.20M$. Between $0.89M$ and $1.19M$ three facilities are opened for some of the instances and for $0.88M$ three facilities are opened for all instances. Evidently even more facilities will be opened when we decrease the capacity of facilities further.

For the SMLRP instances, the average cost is always the same independent of the number of scenarios, and Eindhoven and Almere Oostvaarders are opened the most often (this follows from the weak law of large numbers). From 8 scenarios onwards, Eindhoven and Almere Oostvaarders are opened for all 10 instances. However, increasing the number of scenarios still decreases the standard deviation of the optimal objective values. Owing to time and stability reasons (see online Appendix C), we use instances with 16 scenarios from now on. These instances can be solved in approximately 20 minutes. When we drop the infinite capacity assumption and vary the capacities of the facilities, the solution remains Eindhoven and Almere Oostvaarders for all 10 instances when the capacity of all facilities is above $0.85M$. Between $0.78M$ and $0.85M$ Groningen or Maastricht is opened as third facility for some of the instances and for $0.77M$ Groningen or Maastricht is opened as third facility for all instances. Further decreasing the capacity obviously results in more facilities being opened.

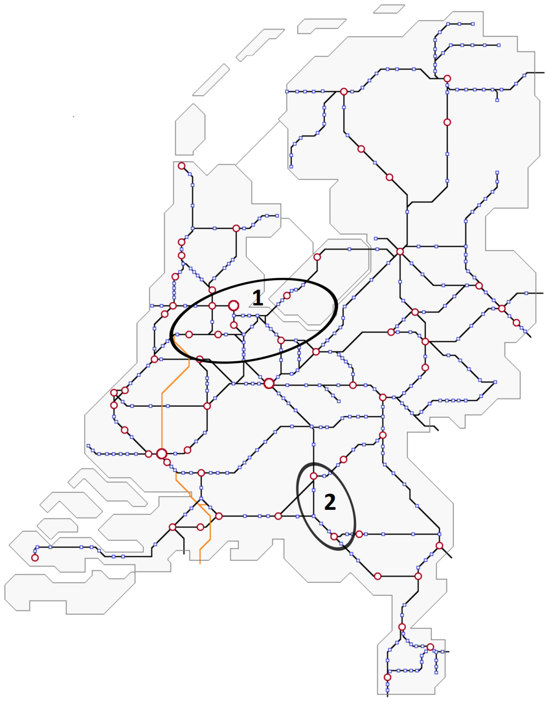



Figure 4: Railway map of the Netherlands with all stations. The larger black circles are the best locations to open a maintenance facility. The small circles are the larger intercity stations and the tiny circles are the smaller regional stations. This picture is adapted from `https://upload.wikimedia.org/wikipedia/commons/6/6a/Spoorkaart_Nederland%2C_IC_stations.png`.

#### 7.2.1 The benefit of the network interchange budget

We make 10 instances with 16 (SMLRP) or 8192 (RMLRP) scenarios according to the plan described in Section 7.1. For each of these instances, the budget is $G^d = 0, 0.25M, 0.5M, 0.75M, M, 2M$, and $\infty$ for all scenarios. Note that a budget of $G^d = 0$ for all scenarios corresponds to a model where maintenance routing is not possible. Furthermore, we include 10 instances where the station interchange restrictions $g_e$ are removed, while also having an infinite network interchange budget. These instances demonstrate the maximum gain that can be achieved by improving the station interchange budgets.

Table 7 shows the influence of the annual network interchange for the SMLRP. Here #sols denotes the number of different facility location decisions among the 10 problem instances. The next column gives the facility location decision that occurs most often and the last column shows the average costs (optimal objective value) in million euros per year. Two of the three earlier mentioned solutions return in Table 7. Starting from a network interchange budget of $0.50M$, the solution is almost always Eindhoven and Almere Oostvaarders. The interchange budget is not binding for $G^d \geq M \ \forall d \in D$ and including maintenance routing can decreases the cost by a total of 25.4%. Removing the interchange station budget decreases the cost with another 13.6%.

| Budget | #sols | Opened facilities | Cost (millions) |
|---|---|---|---|
| 0 | 2 | 's-Hertogenbosch and Hoofddorp (6/10) | 5.9 |
| $0.25M$ | 3 | Eindhoven and Hilversum (5/10) | 5.2 |
| $0.50M$ | 2 | Eindhoven and Almere Oostvaarders (9/10) | 4.7 |
| $0.75M$ | 1 | Eindhoven and Almere Oostvaarders (10/10) | 4.5 |
| $M$ | 1 | Eindhoven and Almere Oostvaarders (10/10) | 4.4 |
| $2M$ | 1 | Eindhoven and Almere Oostvaarders (10/10) | 4.4 |
| $\infty$ | 1 | Eindhoven and Almere Oostvaarders (10/10) | 4.4 |
| No $g_s$ | 3 | Eindhoven and Almere Oostvaarders (8/10) | 3.8 |

Table 7: Network interchange budget results for the SMLRP.

Table 8 shows the influence of the annual network interchange budget for the RMLRP. Again, all solutions consist of one maintenance facility in the south of the large Randstad and one in the north. 's-Hertogenbosch and Hoofddorp may be preferable for instances with a low network interchange budget, while Eindhoven and Hilversum may be preferable for instances with a high network interchange budget. In almost all cases, two facilities are opened, one in the north of the Randstad and one at the south of the large Randstad. The network interchange budget is not binding for $G^d \geq 0.75M \ \forall d \in D$ and increasing the budget decreases the cost by 15.7%. When we remove the station interchange restrictions, we can decrease the cost by another 10.2%. Furthermore, the RMLRP solution values are higher than the solution values of the SMLRP, as expected.

### 7.2.2 Expected value of perfect information and stochastic/robust solution

In the initial experiments, the combination of two facilities – one in the north and one in the south of the randstad in respectively circle 1 and 2 of Figure 4 – provide a good solution. The optimality gaps of these solutions are usually around 2%. Similar solutions are found in Section 7.2.1 and online Appendix C independent of the network interchange budget or the number of scenarios.

We will now look at the expected value of perfect information, and the value of the stochastic or robust solution. Both measures are common measures within the stochastic programming literature (Birge and Louveaux, 1997). The expected value of perfect information is the value of the SMLRP/RMLRP minus

| Budget | #sols | Opened facilities | Cost (millions) |
|---|---|---|---|
| 0 | 2 | ’s-Hertogenbosch and Hoofddorp (8/10) | 7.0 |
| $0.25M$ | 4 | ’s-Hertogenbosch and Hoofddorp (6/10) | 6.3 |
| $0.50M$ | 4 | Eindhoven and Hilversum (5/10) | 6.0 |
| $0.75M$ | 3 | ’s-Hertogenbosch and Hoofddorp (6/10) | 5.9 |
| $M$ | 7 | Eindhoven and Hilversum (4/10) | 5.9 |
| $2M$ | 6 | Eindhoven and Hilversum (4/10) | 5.9 |
| $\infty$ | 6 | Eindhoven and Hilversum (4/10) | 5.9 |
| No $g_s$ | 4 | ’s-Hertogenbosch and The Hague (8/10) | 5.3 |

Table 8: Network interchange budget results for the RMLRP.

the wait and see solution. It is a measure that assesses how valuable perfect information about the future is. The value of the stochastic and robust solution is the improvement of adding future scenarios compared with only using the current situation. The expected value for the current case can be evaluated by solving the second-stage problem for all scenarios while using the opened facilities of the current situation. The value of the robust/stochastic solution is now the expected value of the current case minus the SMLRP/RMLRP optimal objective value.

When we solve the current case scenario with a network interchange budget of $0.75M$, the optimal solution is to open only Almere Oostvaarders with an optimal objective of 3.3 million per year. However, when we evaluate this solution by solving the MRP for an instance with 65536 scenarios, the objective becomes 5.4 million euros per year for the average case objective and 7.7 million euros per year for the absolute robust objective. As a consequence, the expected value of the stochastic/robust solution is respectively 9.3% and 18.2%. This means that neglecting uncertainty about future line plans costs on the order of 10%. The optimal objective for the wait and see solution (8192 scenarios), a solution for which the best facilities are opened for each scenario separately, is 4.6 million for the SMLRP and 6.1 million for the RMLRP. This gives an expected value of perfect information of 2.1% for the SMLRP and 3.2% for the RMLRP.

We expect that the expected value of perfect information is low, because the combination of opening a maintenance facility in the north and south Randstad (see Figure 4) is a good solution for all scenarios. When the solution is not optimal, it is generally close to optimal. For the NS this implies three important insights:

1. It is important to model uncertainty about the future to make good decisions. Ignoring this uncertainly increases cost by around 10%.

2. The risk attitude of the decision maker hardly affects the best decisions, although it does affect the objective value. In particular, any solution with a facility in circles 1 and 2 of Figure 4 is close

to optimal.

3. Investing in more accurate forecasts of future line plans (expected value of perfect information) has very limited benefits.

# 8 Conclusion

We have formulated two novel models for the maintenance location routing problem and used different algorithms to solve them. Our IMIP formulation and the SA method work quite well and can solve industrial size instances. The SA method performs computationally better than the IMIP for the RMLRP and requires less memory as it only has to consider a subset of the scenarios. We expect that the success of our IMIP formulation can be explained by the fact that it is a very efficient formulation that simplifies a multi-commodity problem to a single-commodity problem and reduces the memory requirements by decreasing the number of variables and constraints by $\Theta(\sum_{d\in D}\left(|L^d|^2|N_C| + |L^d||A_I^d|\right))$ and $\Theta(\sum_{d\in D}\left(|L^d|^2 + |N_C||L^d|\right))$, respectively. Furthermore, the solution speed of MIP solvers have increased enormously in recent years, while the current 64-bit computer architectures has increased the maximum RAM usage from $2^{32}$ different values (3-4 GB) to $2^{64}$ values (18 EB, 1 EB = $10^9$ GB).

The case study at the NS indicates that a robust solution exists that gives good values for both the SMLRP as RMLRP. This solution consists of opening two facilities one in the north and one in the south of the larger randstad (circles 1 and 2 of Figure 4). Furthermore, cost savings up to 25.4% for the SMLRP and up to 15.7% for the RMLRP can be made by improving the network interchange budget and including scenarios saves 9.3% for the SMLRP and 18.2% for the RMLRP relative to only using the current line plan scenario.

An interesting future direction for research is to make the price of the facilities dependent on the provided capacity. In such a case, solutions with (approximately) the same facility costs, can consist of many small facilities or a few larger facilities. As it is costly to have sufficient capacity for all possible scenarios, even unlikely ones, we can also allow some recoverability. This recoverability can consist of building additional facilities, selling facilities, or upgrading them to a higher capacity.


# Acknowledgements

The authors thank the Associate Editor and the three anonymous referees for suggestions that improved the paper. In addition the authors would like to thank the people from the Netherlands Railways for the many discussions. We want to especially acknowledge Mei Li Kho, for gathering the current case (2015) data for the case study, and Bob Huisman and Geert-Jan van Houtum for giving valuable input. This study was funded by Nedtrain. Joachim Arts gratefully acknowledges funding through veni grant number 451-16-025.

# References


E. Álvarez-Miranda, E. Fernández, and I. Ljubić. The recoverable robust facility location problem. *Transportation Research Part B: Methodological*, 79:93–120, 2015.

Y. An and B. Zeng. Exploring the modeling capacity of two-stage robust optimization: Variants of robust unit commitment model. *IEEE Transactions on Power Systems*, 30(1):109–122, 2015.

Y. An, B. Zeng, Y. Zhang, and L. Zhao. Reliable p-median facility location problem: two-stage robust models and algorithms. *Transportation Research Part B: Methodological*, 64:54–72, 2014.

L. Anderegg, S. Eidenbenz, M. Gantenbein, C. Stamm, D. S. Taylor, B. Weber, and P. Widmayer. Train routing algorithms: Concepts, design choices, and practical considerations. In *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments*, volume 111, pages 106–118. SIAM, 2003.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, 2013.

J. R. Birge and F. V. Louveaux. *Introduction to Stochastic Programming.* Springer, 1997.

T. C. Chan, Z.-J. M. Shen, and A. Siddiq. Robust defibrillator deployment under cardiac arrest location uncertainty via row-and-column generation. *Operations Research*, 2017.

L. Clarke, E. Johnson, G. Nemhauser, and Z. Zhu. The aircraft rotation problem. *Annals of Operations Research*, 69:33–46, 1997.

A. M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429–1450, 2005.

Z. Drezner and H. W. Hamacher. *Facility location: applications and theory.* Springer Science & Business Media, 2001.

T. A. Feo and J. F. Bard. Flight scheduling and maintenance base planning. *Management Science*, 35 (12):1415–1432, 1989.

V. Gabrel, M. Lacroix, C. Murat, and N. Remli. Robust location transportation problems under uncertain demands. *Discrete Applied Mathematics*, 164:100–111, 2014.

R. Gopalan. The aircraft maintenance base location problem. *European Journal of Operational Research*, 236(2):634–642, 2014.

R. Jiang, M. Zhang, G. Li, and Y. Guan. Benders decomposition for the two-stage security constrained robust unit commitment problem. *Online Optimization*, 2011.

M. Khatami, M. Mahootchi, and R. Z. Farahani. Benders decomposition for concurrent redesign of forward and closed-loop supply chain network with demand and return uncertainties. *Transportation Research Part E: Logistics and Transportation Review*, 79:1–21, 2015.

P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

G. Laporte, S. Nickel, and F. S. da Gama. *Location science*. Springer, 2016.

C. Lee, C. Liu, S. Mehrotra, and Z. Bie. Robust distribution network reconfiguration. *IEEE Transactions on Smart Grid*, 6(2):836–842, 2015.

K. T. Lieckens, P. J. Colen, and M. R. Lambrecht. Optimization of a stochastic remanufacturing network with an exchange option. *Decision Support Systems*, 54:1548–1557, 2013.

G. Maróti and L. Kroon. Maintenance routing for train units: The transition model. *Transportation Science*, 39(4):518–525, 2005.

G. Maróti and L. Kroon. Maintenance routing for train units: The interchange model. *Computers & Operations Research*, 34(4):1121–1140, 2007.

M. T. Melo, S. Nickel, and F. Saldanha-da Gama. Facility location and supply chain management–a review. *European Journal of Operational Research*, 196(2):401–412, 2009.

G. Nagy and S. Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

F. Piu and M. G. Speranza. The locomotive assignment problem: a survey on optimization models. *International Transactions in Operational Research*, 21(3):327–352, 2014.

Prorail and NS. Beter en Meer, 2014. URL `https://www.rijksoverheid.nl/documenten/rapporten/2014/03/28/bijlage-1e-operationele-uitwerking-prorail-en-ns-beter-en-meer`.

J. A. Rappold and B. D. Van Roo. Designing multi-echelon service parts networks with finite repair capacity. *European Journal of Operational Research*, 199:781–792, 2009.

E. D. Santibanez-Gonzalez and A. Diabat. Solving a reverse supply chain design problem by improved Benders decomposition schemes. *Computers & Industrial Engineering*, 66(4):889–898, 2013.

T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro. A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115, 2005.

A. Sarac, R. Batta, and C. M. Rump. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research*, 175(3):1850–1869, 2006.

Z.-J. M. Shen, R. L. Zhan, and J. Zhang. The reliable facility location problem: Formulations, heuristics, and approximation algorithms. *INFORMS Journal on Computing*, 23(3):470–482, 2011.

L. V. Snyder. Facility location under uncertainty: a review. *IIE Transactions*, 38(7):547–564, 2006.

C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *ACM SIGACT News*, 37(1):33–46, 2006.

K. T. Talluri. The four-day aircraft maintenance routing problem. *Transportation Science*, 32(1):43–53, 1998.

H. Üster and H. Agrahari. A Benders decomposition approach for a distribution network design problem with consolidation and capacity considerations. *Operations Research Letters*, 39(2):138–143, 2011.

J. C. W. van Ommeren and A. F. Bumb. Locating repair shops in a stochastic environment. *Computers & Operations Research*, 33:1575–1594, 2006.

W. Xie, K. Somani, et al. Optimizing location and capacity for multiple types of locomotive maintenance shops. *Computer-Aided Civil and Infrastructure Engineering*, 31(3):163–175, 2016.

B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

L. Zhao and B. Zeng. Robust unit commitment problem with demand response and wind energy. In *2012 IEEE power and energy society general meeting*, pages 1–8. IEEE, 2012.