Erasmus University Rotterdam

Erasmus School of Economics

Master's Thesis

Econometrics and Management Science - Operations Research & Quantitative Logistics

Nonlinear Variation Constraints in Railway Crew Scheduling

Author:
M.S. Meijer, BSc
384271

Supervisors:
Prof. Dr. D. Huisman (ESE,NS)
Ir. W. van Dijk (NS)

Co-reader:
Dr. P.C. Bouman (ESE)

August 27, 2017

Abstract

Crew scheduling at railway companies is challenging, since multiple conditions need to be taken into account. The crew scheduling algorithm currently in use at the Netherlands Railways facilitates this task. However, it is difficult to incorporate nonlinear restrictions on variation in routes of crew members.

This thesis describes a study to incorporate such nonlinear restrictions. The aim of this study is twofold. First, it is investigated how the formulation of the crew scheduling problem can be extended to include nonlinear variation constraints. Second, the solution method is adjusted.

Two different formulations for the given problem are proposed. In the first formulation, the nonlinear variation measure is linearized exactly. Since this formulation was expected to be difficult to solve, another formulation is presented in which the variation measure is approximated. Both formulations were used to extend the current crew scheduling algorithm. Since violations of the variation restrictions are penalized in all algorithms, we were able to compare both developed methods to the current algorithm that does not take into account variation.

Two types of test instances were used to evaluate the performance of the methods. First, artificial instances of different sizes were used. Next, we used single- and multi-day instances based on real life data.

On the artificial instances, the first method, in which the variation measure is linearized exactly, is able to generate schedules with more variation than those obtained using the algorithm that does not include variation restrictions, with limited additional duties. Running times and objective values, consisting of duty costs and penalty costs, are comparable to the algorithm without incorporating variation restrictions. The second method yields good solutions on artificial instances, although with slightly higher costs than the first method.

On the single-day real life instance, the performance of the first method is similar to that on the artificial instances. When applying this method instead of the one that does not incorporate variation, penalty costs can be decreased by 15 to 100 percent. Since up to 5 percent more duties are selected, duty costs also increase with up to 5 percent. As a consequence, objective values are similar to those of the algorithm that does not incorporate variation. Furthermore, required running time for this method was lower than for the method that does not consider variation. Running time of the second method turned out to be too large for this method to be of practical value on the real life instance.

Finally, we used a multi-day real life instance, containing tasks for multiple weekdays. On this instance, the method based on exact linearization of the variation restrictions clearly outperformed the basic model that does not incorporate variation. Where the solution of the basic algorithm only contained slightly more variation than for the single-day instance, the first method was able to increase variation by nearly 80 percent. In addition, running time was lower when we incorporated the variation restriction in comparison to when we did not.

In conclusion, both methods are able to solve problems with variation restrictions, but running time of the second method becomes a limiting factor. Since the exact linearization method can increase variation in the schedule with a limited effect on running time or costs, this provides an interesting opportunity to directly include nonlinear variation restrictions in the crew scheduling algorithm at the Netherlands Railways.

Keywords: Crew Scheduling, Netherlands Railways, Lagrangian Relaxation, Set Covering Model, Nonlinear Constraints

Acknowledgements

This thesis was written in fulfillment of the master program Econometrics and Management Science, with specialization Operations Research and Quantitative Logistics, at the Erasmus School of Economics in Rotterdam. During the course of this study, I did an internship at the innovation department π at Netherlands Railways.

I would like to express my gratitude to all the people I cooperated with during my studies at the Erasmus University, both lecturers and my fellow students. However, there are some people I would especially like to thank with respect to this thesis project.

First of all, I would like to thank professor Dennis Huisman, my supervisor from the Erasmus School of Economics. Thanks to his help in finding a suitable project I was able to start my internship at NS. In addition, he advised me during the course of this project and offered his feedback. This was very helpful for improving my methods and bringing this thesis to a higher level. I also want to thank the second reader dr. Paul Bouman for his time to read and evaluate this thesis.

I would also like to thank the staff at π for giving me the opportunity to do my internship at their site and for the valuable experiences. Especially, I would like to thank Wilmet van Dijk, my supervisor at NS, for guiding me during this process. Thanks to our meetings I gained understanding in the practical implications of the considered variation restrictions and of the crew scheduling process at NS in general.

Mirjam Meijer, August 2017

Contents

List of Tables

1 Introduction

Netherlands Railways (Nederlandse Spoorwegen, NS) is the largest passenger railway operator in the Netherlands. Every day around 5,000 scheduled trains are operated that carry over one million passengers on regular working days (Abbink et al., 2005). All operated trains require a train driver and one or more guards. These crew members are based at 28 depots, from which they operate. Train drivers and guards can be relieved at all major stations on a route. A task is the smallest piece of work that can be assigned to one crew member. Therefore, every route is split into several tasks, each consisting of the work on a route between two relief points. This means that on a regular working day around 15,000 tasks need to be assigned to train drivers and around 18,000 to guards, since certain trains require multiple guards (Abbink et al., 2011). These tasks need to be combined into duties that comprise the job list for one anonymous crew member on a single day. During this process many workforce constraints need to be taken into account, for example concerning duration of a duty and break regulations.

As a consequence, crew scheduling is a challenging problem. Currently, NS is using an algorithm for crew scheduling called LUCIA. This method uses a set covering formulation in which duties need to be selected such that all tasks are covered. Since the set of possible duties can grow extremely large, column generation is applied. Columns, i.e. the set of duties from which we can select, are generated using a restricted shortest path problem on a network where nodes represent tasks that need to be performed. The set covering problem is solved using a heuristic method that applies Lagrangian relaxation and subgradient optimization (Abbink, 2014).

The current crew scheduling procedure aims at finding an optimal allocation of tasks to duties. This means that costs are minimized while workforce constraints are satisfied. However, this approach normally does not take into account preferences of employees, concerning for example train types or routes. Division of tasks among crew members has therefore often led to tension. This is the case because some routes are known to be more aggression sensitive than others and intercity connections are often preferred over regional trains. Furthermore, train drivers and guards prefer to have variation in their schedules instead of going back and forth on the same line. To facilitate this process several rules have been agreed upon to create a more even distribution of "sweet and sour" over the employees. The present study focuses on incorporating one of the rules regarding employees' preferences with respect to variation in routes in the crew scheduling algorithm, as opposed to manually steering the algorithm towards a good solution with respect to variation.

The aim of this study is to investigate if it is possible to include the norm on route variation in the current crew scheduling model and how the resulting model can be solved. First, we need a clear definition of variation. Variation is defined as a number of unique kilometers. Since depots differ significantly in size, we measure the average number of unique kilometers for a crew member based at a depot instead of the number of unique kilometers assigned to the depot. In this way, it is easier to compare variation over the different crew depots. It is assumed that if a piece of infrastructure, which is a small part of the railway network, is included in five duties per week for a certain crew depot, 25 crew members will encounter this piece in their schedule. Each of these crew members on average will work on this piece once every five weeks. Of course, the number of crew members working on this piece of infrastructure cannot exceed the number of available crew members at the depot. Therefore, the number of crew members that can cover a certain piece of infrastructure is bounded from above. This cutoff in combination with the fact that a piece of infrastructure only counts every fifth time a duty containing this piece is assigned to the depot makes the variation measure highly nonlinear. Therefore, incorporating the variation norm in the currently available crew scheduling algorithms is not straightforward.

This thesis is outlined as follows. A detailed description of the problem is presented in Section 2, after which an overview of relevant literature is presented in Section 3. Furthermore, in Section 4 we describe how we incorporate variation norms in the formulation of the crew scheduling problem and in Section 5 we discuss how the crew scheduling algorithm is adjusted. In Section 6 test instances are described that are used for testing these methods. Results are described in Section 7, followed by our conclusions in Section 8.

2 Problem Definition

2.1 Planning at NS

Planning processes at NS can be divided into four categories: strategic planning, tactical planning, operational planning and operational control (Abbink, 2014). Strategic planning concerns long term decisions, such as line system planning, in which origins, destinations and frequencies of train lines are decided. Furthermore, decisions are made regarding capacities of crew depots and fleet management, which concerns purchasing new rolling stock. In the tactical planning phase rolling stock is assigned to different lines and a plan is made concerning the pattern for generic days. Based on these generic plans, an operational plan for each calendar day is created in the operational planning phase. In the operational control phase unexpected changes are handled at short notice.

Crew scheduling is part of the tactical planning phase. First, the train timetable is created, which specifies planned departure and arrival times for every train and corresponding stations. The next step is rolling-stock scheduling. In this step it is decided how many units of which rolling stock type are required on different lines. Then, an assignment of specific rolling stock units is made. The final steps of the tactical planning phase concern crew scheduling and rostering. In the crew scheduling phase, duties starting and ending at a certain crew depot are created that describe the tasks that need to be performed by a single crew member on a given day. In the crew rostering step, these duties are assigned to specific crew members.

2.2 Crew Scheduling at NS

Since NS operates a large number of trains every day and employs around 2,700 drivers and 3,000 guards spread over 28 depots, crew scheduling is a complex problem (Abbink, 2014). In addition, duties need to satisfy many constraints concerning labor regulations, such as meal breaks and duty lengths, but also practical regulations regarding knowledge of the type of rolling stock assigned to a task and transfer times between consecutive tasks.

A task is defined to be the smallest amount of work to be assigned to one driver or guard. A duty is the work for one anonymous crew member of a certain depot on a specific day (Abbink et al., 2011). The crew scheduling problem consists of assigning tasks to duties, such that all tasks are covered, taking into account constraints concerning individual duties and the entire set of selected duties. Since many regulations concern an entire week, the problem is solved on a weekly basis. The crew scheduling problem is formulated as a set covering problem, where a duty that needs to be assigned to a crew member is represented by a column. The columns need to be selected in such a way that all tasks are covered. Over-covering is allowed, since it is possible to let a driver or guard be a passenger on a train. In practice, over-covering is limited, since it represents employees that need to be paid, but do not actually execute tasks. However, in some cases over-covering is useful, to increase efficiency of duties, or necessary, if incoming and outgoing tasks at a relief point are unbalanced (Abbink, 2014).

Let S denote the set of all crew depots. For every depot $s \in S$, J_s denotes the set of duties that could be assigned to crew depot s. Let $J = \cup_{s \in S} J_s$ denote the entire set of available duties. The costs corresponding to duty j at depot s are denoted by c_{sj}. Let T denote the set of all tasks that need to be covered. Binary parameter a_{sjt} indicates whether duty $j \in J_s$ includes task t ($a_{sjt} = 1$) or not ($a_{sjt} = 0$). Let J_t denote the set of duties covering task t and T_j the set of tasks covered by duty j. Binary decision variable x_{sj} indicates whether duty $j \in J_s$ is selected in the schedule or not. Finally, let K denote the set of additional constraints that need to be considered and let L_k and U_k denote the corresponding lower and upper bounds and g_{sjk} the coefficient of duty $j \in J_s$ for constraint $k \in K$. The following formulation is used for the basic crew scheduling problem:

$$\min \quad \sum_{s\in S}\sum_{j\in J_s} c_{sj}x_{sj} \tag{2.1}$$

$$\text{s.t.} \quad \sum_{s\in S}\sum_{j\in J_s} a_{sjt}x_{sj} \geq 1 \qquad \forall t \in T \tag{2.2}$$

$$L_k \leq \sum_{s\in S}\sum_{j\in J_s^*} g_{sjk}x_{sj} \leq U_k \qquad \forall k \in K \tag{2.3}$$

$$x_{sj} \in \{0,1\} \qquad \forall s \in S, \forall j \in J_s \tag{2.4}$$

Objective (2.1) is to minimize total costs of all selected duties. Constraints (2.2) require that every task is covered by at least one duty. Constraints (2.3) take into account additional constraints concerning the entire set of selected duties or a subset thereof, denoted by J_s^* for depot s. Finally, Constraints (2.4) enforce that the decision variables are binary, i.e. a duty is either selected or not.

2.2.1 Crew Scheduling Algorithm at NS: LUCIA

The crew scheduling algorithm applied at NS, described by Abbink (2014), iterates between solving the master problem, given by Equations (2.1) to (2.4), for a given set of duties and generating new duties. At the start the algorithm generates an initial set of duties consisting of infeasible duties covering one task. The costs of these duties are set high, such that they are only selected in the final solution in case no feasible solution to the problem exists. In the main algorithm, given in Algorithm 1, the set covering problem is solved, after which uninteresting columns (duties) are deleted and new columns are generated.

Algorithm 1 Overview LUCIA

```
Initialization: sols = ∅ and fixedCols = ∅
while UB - LB ≥ 1 AND #iterations ≤ 300 do
    Solve master problem
    Choose columns to delete
    while #subiterations ≤ 10 do
        Generate columns
        Add generated columns to master problem
        if #generated columns < 5 then
            Exit while loop
        end if
    end while
end while
```

The master problem is solved using a heuristic method, described by Caprara et al. (1999), that applies Lagrangian relaxation and subgradient optimization. New columns are generated using a resource constrained shortest path problem as discussed by Huisman (2007). We will elaborate on these methods in Section 3.

2.3 Variation Norm: Unique Kilometers

One of the proposals agreed upon by NS and the workers council for establishing a fair division of "sweet and sour" concerns the variation in the work of a train driver or guard. This is an important factor for employee satisfaction. Two factors are of importance when modeling this, namely the way in which variation is measured and the amount of variation that is required.

Variation can be measured by the number of unique kilometers assigned to a depot. For this, it is of importance how often pieces of infrastructure, defined as a part of the railway network between two endpoints, occur in duties of the considered depot and how many crew members it hosts. If many crew members are based at a depot, but a piece of infrastructure only occurs in a few duties, this piece can only be covered by a limited number of crew members. If we consider a different depot with fewer crew members, but more assigned duties covering this piece of infrastructure, more of these crew members will be working on this piece of infrastructure. Hence, since depots have varying sizes and the number of duties in which a piece occurs might be different, the number of unique kilometers per depot does not have much value. Therefore, the number of unique kilometers per train driver or guard is used, instead of the number of unique kilometers per depot. However, since it is not yet known which crew member at the depot will perform which tasks, this value is unknown. Therefore, we estimate this value based on the duties that are assigned to the crew depot. This estimation is based on two assumptions. Firstly, per five duties in a week in which a piece of infrastructure occurs it can be included in the work package of 25 crew members. Each crew member will then on average encounter the piece once every five weeks. Secondly, it is assumed that the division of duties among employees of a depot takes into account variation. This means that it is assumed that the possibilities for variation created in the crew scheduling phase are utilized in the crew rostering phase.

The number of unique kilometers per train driver or guard is computed as follows. Let m_s be the number of train drivers or guards for depot $s \in S$. Denote by $i \in I$ one of the pieces of infrastructure and by b_{is} and k_i the number of duties for depot s in which i occurs and the length of i in kilometers, respectively. The number of unique kilometers per train driver or guard for depot s is then given by Equation (2.5).

$$\sum_{i \in I} \frac{k_i \cdot \min\left\{m_s, 25\left\lfloor \frac{b_{is}}{5} \right\rfloor\right\}.}{m_s} \tag{2.5}$$

Consider, for example, a crew depot with 100 train drivers. A certain piece of infrastructure has a length of 10 kilometers and occurs in 12 duties during a week. This piece can be included in the schedule of $25 \cdot \left\lfloor \frac{12}{5} \right\rfloor = 50$ train drivers. Hence, on average it adds 5 kilometers to the number of unique kilometers per train driver. If there would be 30 train drivers based at this crew depot, the same piece of infrastructure would occur in every schedule, since $30 < 50$, and therefore it would add 10 kilometers to the average number of unique kilometers per driver. Hence, both the number of duties in the schedule covering a certain piece of infrastructure for a specific depot and the number of crew members based at this depot influence the contribution of this piece of infrastructure to the number of unique kilometers per train driver.

On average, a train driver is assigned 3.2 duties per week. Therefore, the number of train drivers at depot s can be estimated by the number of selected duties for depot s in a week divided by 3.2, $m_s = \frac{\sum_{j \in J_s} x_{sj}}{3.2}$. Furthermore, the value of b_{is} depends on which duties are selected, hence b_{is} can be expressed in terms of x_{sj}. The value of b_{is} is equal to the number of duties for crew depot s that are selected and have at least one task covering infrastructure piece

i. Therefore, parameter h_{sji} is introduced, which equals 1 if duty $j \in J_s$ contains infrastructure piece i. Let f_{ti} be a binary parameter, indicating whether task t contains infrastructure piece i. Hence, $a_{sjt}f_{ti} = 1$ if duty $j \in J_s$ contains task t that makes use of infrastructure piece i. Then h_{sji} can be defined in the following way

$$h_{sji} = 1 - \prod_{t \in T}(1 - a_{sjt}f_{ti}) \tag{2.6}$$

The term inside the product equals 0 if task t is in duty $j \in J_s$ and infrastructure piece i is covered in task t. Hence, the product itself equals 0 if this holds true for at least one task. Now, b_{is} can be written in the following way

$$b_{is} = \sum_{j \in J_s} x_{sj}h_{sji} \tag{2.7}$$

These expressions can now be inserted in the measure for variation, to obtain an expression in terms of x_{sj}, leading to the following variation measure.

$$\sum_{i \in I} \frac{k_i \cdot \min\left\{\frac{\sum_{j \in J_s} x_{sj}}{3.2}, 25\left\lfloor\frac{\sum_{j \in J_s} x_{sj}h_{sji}}{5}\right\rfloor\right\}}{\frac{\sum_{j \in J_s} x_{sj}}{3.2}} \tag{2.8}$$

The second factor that is of importance is the norm value. The norm value for variation used by NS is different for train drivers and guards. Furthermore, there is a requirement per depot and an average national requirement. Table 2.1 gives the variation norms for the average number of unique kilometers over all crew members and for different depots for both train drivers and guards. For the number of unique kilometers per depot, two numbers are given for both train drivers and guards. The first number for the depots is the minimum number that should hold for most depots. However, this may be violated for a limited number of depots which together host at most ten percent of the total number of train drivers or guards. The number between brackets is the minimum for these depots. These exceptions are necessary, since it is sometimes impossible to reach the first number due to the location of some depots. Since the developed methods will be suitable for different datasets and norm values, we will use more general notation for these norm values. Hence, let V_s^H denote the high norm value and let V_s^L be the low norm value. In addition, we denote the average norm value by G.

Table 2.1: Variation norms for train drivers and guards

	Train driver	Guard
Average number of unique kilometers in total	850	950
Number of unique kilometers per depot	700 (550)	725 (600)

Since the considered constraints do not concern individual duties, but the complete allocation of duties, they should be incorporated in the problem formulation described in Section 2.2. However, since the constraints are highly nonlinear it is not possible to directly add them as additional constraints to the set covering formulation. Therefore, these constraints should be linearized or modeled differently. In this study we investigate how these constraints can be modeled and how the resulting model can be solved. Since the constraints are formulated in the same way for both train drivers and guards, the same models can be applied to both cases. However, in this study, we only test the model using test instances containing tasks for train drivers.

2.4 Aim of the Thesis

Currently, the variation restrictions are taken into account by manually steering the solution to include certain duties, such that the variation restrictions are satisfied. However, ideally, these restrictions would be included directly into the crew scheduling algorithm, such that resulting schedules adhere to the constraints.

The present thesis will focus on how to adjust the solution approach for the master problem, such that the resulting schedules satisfy the variation restrictions. The aim is to 1) determine whether it is possible to incorporate constraints regarding average variation per train driver or guard into the formulation of the crew scheduling problem, and 2) to investigate whether it is possible to adjust the current crew scheduling algorithm such that these constraints are taken into account directly, rather than having to adjust the solution afterwards, and determine whether solutions produced by such model are superior to the basic model, without considering variation restrictions, in terms of satisfaction of the variation norms and overall costs.

3 Literature Review

Over the past decades extensive research has been done on crew scheduling, especially in the airline industry, but also in the railway industry. This has resulted in better and faster solution methods for crew scheduling problems. The variation norm under consideration is very specific for the NS case, but other studies have addressed problems involving other types of nonlinear constraints. First, we will give an overview of literature on solving crew scheduling problems, including some studies on fairness of crew schedules. Next, literature will be presented concerning solution approaches for problems with nonlinear constraints.

3.1 Crew Scheduling

3.1.1 Solving the Crew Scheduling Problem

Crew schedules have to adhere to many constraints, e.g. due to legislation and labor agreements. In order to incorporate these constraints, the problem is often modeled as a set partitioning or set covering problem. In this way, constraints concerning individual duties can be taken into account in the duty generation phase.

Column Generation When the number of tasks to be executed increases, the number of possible duties can become very large. It can therefore be very inefficient or even impossible to create all possible duties and solve the set covering problem using these columns. Therefore, for solving large scale crew scheduling problems, column generation is often applied. The main idea behind this method, which was introduced by Dantzig and Wolfe (1960), is to solve the set covering problem with a small subset of all possible columns only. The method then iterates between solving the so-called master problem and creating new columns by solving the subproblems or pricing problems. When it is not possible to improve the optimal solution of the master problem by including additional columns, the optimal solution of the complete problem has been reached.

In the pricing problem, columns are generated by solving another optimization problem, which is in the case of the crew scheduling problem a resource constrained shortest path problem. Huisman (2007) discusses solving the shortest path problem to create duties, taking into account a meal break constraint. The meal break constraint is not explicitly modeled, but generated paths that do not contain a meal break opportunity are discarded. For this purpose, k shortest paths over the network consisting of the tasks to be executed are generated and the best one that includes a meal break is selected.

Master Problem The solution approach proposed by Caprara et al. (1999) for solving the set covering formulation, i.e. master problem, has been applied in many solution methods. They present a method for solving the set covering problem using Lagrangian relaxation and subgradient optimization. Lagrangian relaxation is a technique that can be applied to obtain bounds on the optimal solution. Difficult constraints can be relaxed and violations of these constraints are penalized. For more information on Lagrangian relaxation consult for example Fisher (1981). The algorithm proposed by Caprara et al. (1999) consists of three phases. In the first phase, a near-optimal vector of Lagrangian multipliers is obtained. Next, a heuristic phase is applied to generate multiple solutions starting from this near-optimal vector of multipliers. In the third phase, column fixing is applied to select 'good' columns that are likely to be part of useful solutions. One of the main contributions of this approach is a pricing method. In all phases a small subset of columns, called the core, is considered. This core is updated using dual information associated with the Lagrangian multipliers, where the columns with lowest Lagrangian cost are included in the core. The use of pricing has a large effect on computation time, especially for large problems. After the algorithm has been applied, a refining procedure is used to improve a close to optimal solution. First, the gap between the lower and upper bound is calculated, after which for each column its contribution to the gap is estimated. Columns with a small contribution to the gap are expected to be part of the optimal solution, hence these columns are fixed. The resulting problem is re-optimized using the three phase procedure. The algorithm terminates when all tasks are covered by the fixed columns or the lower bound is no longer smaller than the best objective value.

Alternative Solution Approaches Other studies discuss methods to reduce the size of the problem and therefore make the problem easier to solve. For example, Jütte and Thonemann (2012) describe the crew scheduling problem of a large railway freight carrier in Germany. To reduce the size of the problem, tasks are assigned to subregions that can be solved separately. To avoid making the problem too restrictive, subregions can overlap and certain tasks can be considered to be part of multiple subregions. It was concluded that this approach could produce interesting results, but their value was highly dependent on the considered division into subproblems. The Dutch railway infrastructure does not allow for a clear division into subregions. Since travel times are relatively short, tasks in the north of the country could be performed by crew members based in the south. Furthermore, dividing the country into subregions would restrict possibilities for variation. Consequently, this approach is considered not to be suitable for the problem considered in the current study.

Kroon and Fischetti (2001) consider a different type of aggregation. They do not consider separate problems for train drivers and guards, but combine them into a team. This also has the advantage that the schedule becomes more robust, since the number of factors that can cause delays is reduced. However, there are some trains that require multiple guards. If this is the case separate duties need to be generated. In addition, Kroon and Fischetti (2001) discuss the conflicting interests between preferences of crew members and robustness of the schedule. From the point of view of robustness, it would be good to keep a certain team of driver and guard on a certain rolling stock. However, this reduces possibilities for variation in the schedule.

Similar to Kroon and Fischetti (2001), Hanafi and Kozan (2014) consider a combination of train driver and guard that need to be scheduled together. However, to solve the crew scheduling problem they apply simulated annealing. After an initial solution is created, the solution is improved using a hybrid constructive simulated annealing heuristic. Swap operations are performed to exchange positions of trip segments. Such operations are only considered on segments of a combination of trips that both originate and terminate at the same relief point. The insert operation consists of moving one trip segment from one duty to another. After exchange the new solution value is calculated. If it leads to improvement, the new solution is accepted. If the solution value deteriorates, the solution is accepted with a certain probability based on the cooling factor. The cooling factor decreases after a certain number of iterations has been performed. It was concluded that the algorithm can produce near-optimal solutions to the crew scheduling problem within reasonable computation times.

Kornilakis and Stamatopoulos (2002) apply genetic algorithms to the crew scheduling problem. The approach consists of two phases. First, duties are generated taking into account constraints. Next, the set covering problem is solved using genetic algorithms. A solution is given by a chromosome, which consists of genes corresponding to duties. If the gene corresponding to a duty is equal to one, the duty is selected, if the gene equals zero it is not selected. Based on a certain fitness function, 'parents' are selected from which new solutions are created. If a gene has value one (respectively zero) in both parents the gene in the off-spring is also set to one (respectively zero). If both parents have a different value for the gene, the offspring gene value is selected randomly. Afterwards some genes are randomly mutated from one to zero or the other way around to avoid local optima. It may be necessary to correct the solution if not all tasks are covered in a given solution. It was concluded that after a large number of iterations good solutions could be obtained.

3.1.2 Incorporating Crew Preferences

Abbink (2014) describes incorporating several "sharing sweet and sour" rules at NS. Additional constraints are considered to obtain a more equal distribution of agreeable and less agreeable work over the employees. For example, rules concerning standard deviation on the risk of encountering aggression are considered, stating that standard deviation of the percentage aggression work over the different depots should not exceed a certain bound. Similarly, constraints concerning standard deviation of the percentage of work on preferred train types are described. Due to their nonlinear structure resulting from calculation of the standard deviation, these constraints are currently not modeled explicitly. After the crew scheduling problem is solved the constraints are checked and if needed additional restrictions are imposed after which the problem is re-optimized.

Fair distribution of unpopular tasks has also been studied by Jütte et al. (2017). This study is based on the case of a large freight carrier that noticed an unfair distribution of early shifts over the different depots. In the basic set covering problem penalty costs are included to avoid early shifts if possible. However, this does not take into account distribution of these shifts over the respective depots. To create a fairer division of these shifts, fairness constraints are added to the problem. Since the number of early shifts is not known a priori, but depends on the allocation of tasks to the shifts, no fixed bounds are considered. Instead, penalty costs are included in case unpopularity for a certain depot exceeds average unpopularity of the schedule. It is concluded that in this way a fairer division of unpopular shifts can be obtained, without affecting overall unpopularity or costs of the entire schedule.

3.2 Nonlinear Constraints

Although literature on nonlinear constraints in crew scheduling is scarce, the problem of solving nonlinear programming problems has been studied extensively. In this section, we discuss literature concerning nonlinear constraints both in crew scheduling and in general. It should however be kept in mind that not only nonlinear constraints but also the actual crew scheduling problem itself are largely complex. Therefore, not all of the available methods for dealing with nonlinear constraints will be suitable for existing crew scheduling algorithms.

3.2.1 Incorporating Nonlinear Constraints in Crew Scheduling

A method for incorporating nonlinear constraints in airline crew scheduling, which is being applied at Air France, was developed by Desaulniers et al. (1997). In this problem pairings of duties need to be created that start and end at the same crew base. The problem is formulated as an integer nonlinear multi-commodity network flow problem with additional resource variables. Nonlinearities are present in both the objective function and part of the constraints. The problem is solved using a branch-and-bound algorithm based on Dantzig-Wolfe decomposition. The master problem becomes a set partitioning problem, while the subproblems are resource constraint shortest path problems. All nonlinearities are incorporated in the subproblem, such that the master problem becomes linear. The difference with the crew scheduling problem at NS is that in the case of NS it is not possible to incorporate the nonlinear variation norm into the subproblems, since the nonlinear constraints concern the entire set of selected duties instead of individual duties. As a consequence, the master problem remains nonlinear.

3.2.2 Incorporating Nonlinear Constraints in General

Costa and Oliveira (2001) investigated methods to solve mixed integer nonlinear programming problems in general. First, a genetic algorithm is considered. Constraints are handled by applying a penalty function. In case a solution does not satisfy certain constraints, the fitness function, indicating the value of the solution, is set equal to the maximum function value of all feasible solutions plus the constraint violations. In this way, an infeasible solution will always have a worse fitness value than the feasible solutions. The second approach considered, concerns evolution strategies. Similar to genetic algorithms, new solutions are generated by mutating parent solutions. Costa and Oliveira (2001) show that especially the evolution strategies approach yields good results, even though both methods require long computation times. Furthermore, both methods appeared to have difficulties finding feasible solutions to highly constrained problems.

Cai et al. (2001) endorse the fact that evolutionary search procedures, such as genetic algorithms, often converge slowly and have problems finding feasible solutions for problems with narrow feasible regions. Therefore, they attempt to combine genetic algorithms with linear programming. Complicating variables are identified, such that the problem becomes linear in case these variables are fixed. The problem is then solved sequentially, varying the values of the complicating variables by a genetic algorithm. The method turned out to work well for larger and complex problems, but does converge very slowly to near-optimal solutions. This approach does not seem to be applicable to the crew scheduling problem at NS, since it is not possible to identify complicating variables such that the problem becomes linear.

4 Mathematical Formulations

In this section, we will extend the current formulation of the crew scheduling problem to include variation restrictions. First, we will discuss how the variation constraints can be incorporated in the formulation. Next, we will discuss some simplifying assumptions, followed by the associated formulations.

4.1 Initial Formulations

4.1.1 Constraints

Since the variation constraints do not only concern individual duties, but in fact the entire selection of duties, these need to be incorporated in the set covering problem. The constraint on variation for a specific crew depot can be obtained by restricting the variation measure to exceed the required norm, which is formulated in Equation (4.1).

$$\sum_{i \in I} \frac{k_i \cdot \min\left\{\frac{\sum_{j \in J_s} x_{sj}}{3.2}, 25\left\lfloor\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5}\right\rfloor\right\}}{\frac{\sum_{j \in J_s} x_{sj}}{3.2}} \geq V_s^H \tag{4.1}$$

However, for a number of depots, which together employ less than 10% of the crew, it is allowed to use a lower norm value. This can be incorporated by including a variable indicating which norm value is used for depot $s \in S$. Let v_s be equal to 1 if variation for crew depot s satisfies the high norm and let v_s be equal to 0 if variation satisfies the lower norm, but not the high norm. Constraints can be introduced to guarantee that the depots for which v_s is equal to 0 together do not employ more than 10% of the crew. The constraint for variation per depot can now be formulated as Equation (4.2), when we add Constraint (4.3)

$$\sum_{i \in I} \frac{k_i \cdot \min\left\{\frac{\sum_{j \in J_s} x_{sj}}{3.2}, 25\left\lfloor\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5}\right\rfloor\right\}}{\frac{\sum_{j \in J_s} x_{sj}}{3.2}} \geq V_s^L + (V_s^H - V_s^L) v_s \tag{4.2}$$

$$\sum_{s \in S} v_s \sum_{j \in J_s} x_{sj} \geq 0.9 \sum_{s \in S} \sum_{j \in J_s} x_{sj} \tag{4.3}$$

Finally, the constraint concerning average variation over all crew depots needs to be formulated. For this, a weighted average over all crew depots is calculated, given by Equation (4.4).

$$\frac{\sum_{s \in S} \sum_{i \in I} k_i \cdot \min\left\{\frac{\sum_{j \in J_s} x_{sj}}{3.2}, 25\left\lfloor\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5}\right\rfloor\right\}}{\sum_{s \in S} \frac{\sum_{j \in J_s} x_{sj}}{3.2}} \geq G \tag{4.4}$$

To be able to include these constraints, the variation measure should be linearized. Multiple terms of the variation measure make the constraints nonlinear, namely the minimization of the two parts and the floor function. To avoid the division by decision variables, both sides of the constraints can be multiplied by the denominator. The minimization function can be avoided by including an additional binary variable, indicating which of the two inputs is smaller, and some constraints to restrict the value of the variable. For the variation norm of the individual depots, this introduces the multiplication of decision variables, namely $v_s \sum_{j \in J_s} x_{sj}$. This multiplication can be replaced by auxiliary variable r_s, which requires adding Constraints (4.5) and (4.6) to the problem. Constraint (4.5) ensures that when v_s is equal to one, r_s is equal to $\sum_{j \in J_s} x_{sj}$, whereas Constraint (4.6) guarantees that r_s equals 0 when v_s does.

$$-M(1 - v_s) \leq \sum_{j \in J_s} x_{sj} - r_s \leq M(1 - v_s) \tag{4.5}$$

$$0 \leq r_s \leq M v_s \tag{4.6}$$

Floor Function Most difficult to linearize is the floor function. Since linearizing this part can lead to addition of difficult constraints, two formulations of the problem are considered. First, we formulate a model in which the floor function is linearized exactly. Second, a different formulation is considered in which the floor function is approximated by replacing $\lfloor x \rfloor$ by $x - 1$. This approach makes the constraints stricter, since $\lfloor x \rfloor > x - 1$. Therefore, this approach is expected to give worse solutions in terms of costs, but it may be easier to solve. In the remainder of this section, we discuss the formulations based on these two approaches. In Sections 5.2 and 5.3, we will describe for both formulations how the resulting model can be solved.

4.1.2 Method 1: Linearize Floor Function

We will first consider the formulation in which the floor function is linearized exactly. In this formulation, we let $q_{is} = \left\lfloor \frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} \right\rfloor$. We can now denote $\sum_{j \in J_s} x_{sj} h_{sji} = 5q_{is} + R$, where R is the remainder of the division, $0 \leq R \leq 4$. The floor function in the variation measure can now be replaced by q_{is}. The variation measure can be denoted by Equation (4.7), when we restrict the value of q_{is} by adding constraint $0 \leq \sum_{j \in J_s} x_{sj} h_{sji} - 5q_{is} \leq 4$.

$$\sum_{i \in I} k_i \cdot \min \left\{ \frac{\sum_{j \in J_s} x_{sj}}{3.2}, 25 q_{is} \right\} \tag{4.7}$$

In order to eliminate the minimum operator, we introduce variables u_{is} for every combination of depot s and infrastructure piece i, such that $u_{is} = 1$ when q_{is} is smaller than $\sum_{j \in J_s} x_{sj}$. This requires adding constraints $\sum_{j \in J_s} x_{sj} - q_{is} \leq M u_{is}$. The variation measure is then given by Equation (4.8)

$$\sum_{i \in I} k_i \cdot \left(\frac{\sum_{j \in J_s} x_{sj}}{3.2} (1 - u_{is}) + 25 q_{is} u_{is} \right) \tag{4.8}$$

However, the expression now contains multiplications of decision variables. Therefore, we introduce variables $w_{is} = q_{is} u_{is}$ and $y_{is} = u_{is} \sum_{j \in J_s} x_{sj}$. The variation measure can then be rewritten as Equation (4.9).

$$\sum_{i \in I} k_i \cdot \left(\frac{1}{3.2} \left(\sum_{j \in J_s} x_{sj} - y_{is} \right) + 25 w_{is} \right) \tag{4.9}$$

The following constraints (Equations (4.10) to (4.13)) need to be added to the problem, based on the same reasoning as (4.5) and (4.6).

$$-M(1 - u_{is}) \leq q_{is} - w_{is} \leq M(1 - u_{is}) \tag{4.10}$$

$$0 \leq w_{is} \leq M u_{is} \tag{4.11}$$

$$-M(1 - u_{is}) \leq \sum_{j \in J_s} x_{sj} - y_{is} \leq M(1 - u_{is}) \tag{4.12}$$

$$0 \leq y_{is} \leq M u_{is} \tag{4.13}$$

To avoid the situation that no feasible solutions can be found, we introduce additional variable z_s for every depot $s \in S$. This variable is subtracted from the right-hand side of the individual variation constraint, such that positive values of this value make the constraint more easily satisfied. However, penalty costs P for this variable are included in the objective function. As a consequence, these variables are only assigned positive values if it is not possible or very costly to satisfy the variation norm for the corresponding depot. Similarly, variable z' is introduced for the average variation constraint. Violations of this constraint are also penalized by P per unit violation.

Finally, we need to define the maximum value of q_{is}. Variable q_{is} represents $\left\lfloor \frac{\sum_{j \in J_s} h_{sji} x_{sj}}{5} \right\rfloor$, of which the numerator is bounded above by the total number of duties for depot s that are available to select from and cover infrastructure piece i. Therefore, the maximum value q_{is} can obtain is $\left\lfloor \frac{\sum_{j \in J_s} h_{sji}}{5} \right\rfloor$, which we denote by q_{is}^{max}.

The crew scheduling problem can now be formulated as follows:

$$\min \quad \sum_{s \in S} \left(\sum_{j \in J_s} c_{sj} x_{sj} + P z_s \right) + P z' \tag{4.14}$$

$$\text{s.t.} \quad \sum_{s \in S} \sum_{j \in J_s} a_{sjt} x_{sj} \geq 1 \qquad \forall t \in T \tag{4.15}$$

$$\sum_{i \in I} k_i \left(\frac{1}{3.2} \left(\sum_{j \in J_s} x_{sj} - y_{is} \right) + 25 w_{is} \right) \geq V_s^L \frac{\sum_{j \in J_s} x_{sj}}{3.2} + (V_s^H - V_s^L) \frac{r_s}{3.2} - z_s \quad \forall s \in S \tag{4.16}$$

$$0 \leq \sum_{j \in J_s} x_{sj} h_{sji} - 5 q_{is} \leq 4 \qquad \forall i \in I, \forall s \in S \tag{4.17}$$

$$\sum_{s \in S} r_s \geq 0.9 \sum_{s \in S} \sum_{j \in J_s} x_{sj} \tag{4.18}$$

$$\sum_{s \in S} \sum_{i \in I} k_i \left(\frac{1}{3.2} \left(\sum_{j \in J_s} x_{sj} - y_{is} \right) + 25 w_{is} \right) \geq G \sum_{s \in S} \frac{\sum_{j \in J_s} x_{sj}}{3.2} - z' \tag{4.19}$$

$$\frac{\sum_{j \in J_s} x_{sj}}{3.2} - 25 q_{is} \leq M u_{is} \qquad \forall i \in I, \forall s \in S \tag{4.20}$$

$$-M(1 - u_{is}) \leq q_{is} - w_{is} \leq M(1 - u_{is}) \qquad \forall i \in I, \forall s \in S \tag{4.21}$$

$$0 \leq w_{is} \leq M u_{is} \qquad \forall i \in I, \forall s \in S \tag{4.22}$$

$$-M(1 - u_{is}) \leq \sum_{j \in J_s} x_{sj} - y_{is} \leq M(1 - u_{is}) \qquad \forall i \in I, \forall s \in S \tag{4.23}$$

$$0 \leq y_{is} \leq M u_{is} \qquad \forall i \in I, \forall s \in S \tag{4.24}$$

$$-M(1 - v_s) \leq \sum_{j \in J_s} x_{sj} - r_s \leq M(1 - v_s) \qquad \forall s \in S \tag{4.25}$$

$$0 \leq r_s \leq M v_s \qquad \forall s \in S \tag{4.26}$$

$$x_{sj} \in \{0, 1\} \qquad \forall s \in S, \forall j \in J_s \tag{4.27}$$

$$q_{is} \in \{0, \dots, q_{is}^{max}\} \qquad \forall i \in I, \forall s \in S \tag{4.28}$$

$$u_{is} \in \{0, 1\} \qquad \forall i \in I, \forall s \in S \tag{4.29}$$

$$v_s \in \{0, 1\} \qquad \forall s \in S \tag{4.30}$$

$$w_{is} \in \{0, \dots, q_{is}^{max}\} \qquad \forall i \in I, \forall s \in S \tag{4.31}$$

$$y_{is} \in \{0, \dots, |J_s|\} \qquad \forall i \in I, \forall s \in S \tag{4.32}$$

$$r_s \in \{0, \dots, |J_s|\} \qquad \forall s \in S \tag{4.33}$$

$$z_s \in \{0, \dots, V_s m_s\} \qquad \forall s \in S \tag{4.34}$$

$$z' \in \{0, \dots, G \sum_{s \in S} m_S\} \tag{4.35}$$

Objective (4.14) is to minimize total costs of all selected duties and penalty costs. Constraints (4.15) are the set covering constraints that require that every task is covered by at least one selected duty. Constraints (4.16) represent the variation norm for the individual depots.

Constraints (4.17) are required for linearizing the floor function. They restrict the remainder of the floor function to be smaller than the denominator of its argument, such that the variables q_{is} are defined correctly. Constraint (4.18) guarantees that the depots for which the high variation norm is satisfied carry together at least 90% of the crew members. The average variation norm over all depots is represented by Constraint (4.19). The remaining constraints are required for linearizing the variation expression or restricting the domains of variables. First, Constraints (4.20) make sure that u_{is} correctly indicates which of the two terms that are minimized is smallest. Constraints (4.21) and (4.22) are used to linearize the multiplication of q_{is} and u_{is} and guarantee that when $u_{is} = 1$ (the second term is smaller), q_{is} and w_{is} are equal. Similarly, Constraints (4.23) and (4.24) guarantee that y_{is} has the correct value and Constraints (4.25) and (4.26) restrict the value of r_s. Finally, Constraints (4.27) to (4.35) restrict domains of the variables.

4.1.3 Method 2: Approximate Floor Function

Next, we will explain how the problem can be formulated when the floor function is replaced by subtracting 1 from its argument, i.e. $\lfloor x \rfloor$ is replaced by $x - 1$. Since we are dealing with a larger than or equal constraint, the constraint is less satisfied when the value of $\min\left\{x, 25 \cdot \left\lfloor \frac{b_{is}}{5} \right\rfloor\right\}$ is lower. Since $25 \left\lfloor \frac{a_i}{5} \right\rfloor > 25 \cdot \left(\frac{b_{is}}{5} - 1\right)$, the variation constraint becomes more restrictive if we replace the actual measure by $\sum_{i \in I} k_i \cdot \min\left\{x, 25\left(\frac{b_{is}}{5} - 1\right)\right\}$. This means that if the more restrictive constraint is satisfied, the original constraint is also satisfied. The variation measure could thus be linearized by replacing $\left\lfloor \frac{b_{is}}{5} \right\rfloor$ by $\frac{b_{is}}{5} - 1$.

The minimization can again be linearized by including an additional binary variable, indicating which part of the expression is smallest. We can then express the measure for variation by Equation (4.36).

$$\sum_{i \in I} k_i \left(\frac{\sum_{j \in J_s} x_{sj}}{3.2} (1 - u_{is}) + 25 \left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1 \right) u_{is} \right) \tag{4.36}$$

However, this expression contains multiplication of two binary variables x_{sj} and u_{is}. This can be replaced by an additional variable w_{sji} that is equal to 1 if both $x_{sj} = 1$ and $u_{is} = 1$ and equal to 0 otherwise. In addition, it contains the multiplication of $\sum_{j \in J_s} x_{sj}$ and u_{is}. Denote $y_{is} = u_{is} \sum_{j \in J_s} x_{sj}$, which is equal to $\sum_{j \in J_s} x_{sj}$ if $u_{is} = 1$ and to 0 otherwise. This leads to the expression given by Equation (4.37).

$$\sum_{i \in I} k_i \left(\frac{1}{3.2} \left(\sum_{j \in J_s} x_{sj} - y_{is} \right) + 25 \left(\frac{\sum_{j \in J_s} w_{sji} h_{sji}}{5} - u_{is} \right) \right) \tag{4.37}$$

Certain constraints need to be added to ensure that the auxiliary variables get the appropriate value. First of all, Constraint (4.38) needs to be added for every combination of depot s and infrastructure piece i to ensure variable u_{is} has the correct value.

$$\frac{\sum_{j \in J_s} x_{sj}}{3.2} - 25 \left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1 \right) < M u_{is} \tag{4.38}$$

In addition, the Constraints (4.39) to (4.41) are required to restrict the value of w_{sji}.

$$w_{sji} \leq x_{sj} \tag{4.39}$$

$$w_{sji} \leq u_{is} \tag{4.40}$$

$$w_{sji} \geq x_{sj} + u_{is} - 1 \tag{4.41}$$

To guarantee y_{is} is equal to $\sum_{j \in J_s} x_{sj}$ when $u_{is} = 1$ and 0 otherwise, we add the Constraints (4.42) and (4.43) .

$$-M(1-u_{is}) \leq \sum_{j \in J_s} x_{sj} - y_{is} \leq M(1-u_{is}) \tag{4.42}$$

$$0 \leq y_{is} \leq Mu_{is} \tag{4.43}$$

Similar to the formulation in Section 4.1.2, additional variables z_s and z' are included to avoid not finding any feasible solutions and corresponding penalty costs are included in the objective function. The crew scheduling problem is then formulated as follows:

$$\min \quad \sum_{s \in S} \left(\sum_{j \in J_s} c_{sj} x_{sj} + P z_s \right) + P z' \tag{4.44}$$

$$\text{s.t.} \quad \sum_{s \in S} \sum_{j \in J_s} a_{sjt} x_{sj} \geq 1 \qquad \forall t \in T \tag{4.45}$$

$$\sum_{i \in I} k_i \left(\frac{1}{3.2} \left(\sum_{j \in J_s} x_{sj} - y_{is} \right) + 25 \left(\frac{\sum_{j \in J_s} w_{sji} h_{sji}}{5} - u_{is} \right) \right) \geq V_s^L \frac{\sum_{j \in J_s} x_{sj}}{3.2} + (V_s^H - V_s^L) \frac{r_s}{3.2} - z_s \quad \forall s \in S \tag{4.46}$$

$$\sum_{s \in S} r_s \geq 0.9 \sum_{s \in S} \sum_{j \in J_s} x_{sj} \tag{4.47}$$

$$\sum_{s \in S} \sum_{i \in I} k_i \left(\frac{1}{3.2} \left(\sum_{j \in J_s} x_{sj} - y_{is} \right) + 25 \left(\frac{\sum_{j \in J_s} w_{sji} h_{sji}}{5} - u_{is} \right) \right) \geq G \sum_{s \in S} \frac{\sum_{j \in J_s} x_{sj}}{3.2} - z' \tag{4.48}$$

$$\frac{\sum_{j \in J_s} x_{sj}}{3.2} - 25 \left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1 \right) \leq M u_{is} \qquad \forall i \in I, \forall s \in S \tag{4.49}$$

$$w_{sji} \leq x_{sj} \qquad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.50}$$

$$w_{sji} \leq u_{is} \qquad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.51}$$

$$w_{sji} \geq x_{sj} + u_{is} - 1 \qquad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.52}$$

$$\sum_{j \in J_s} x_{sj} - y_{is} \leq M(1-u_{is}) \qquad \forall i \in I, \forall s \in S \tag{4.53}$$

$$\sum_{j \in J_s} x_{sj} - y_{is} \geq -M(1-u_{is}) \qquad \forall i \in I, \forall s \in S \tag{4.54}$$

$$y_{is} \leq M u_{is} \qquad \forall i \in I, \forall s \in S \tag{4.55}$$

$$y_{is} \geq 0 \qquad \forall i \in I, \forall s \in S \tag{4.56}$$

$$\sum_{j \in J_s} x_{sj} - r_s \leq M(1-v_s) \qquad \forall s \in S \tag{4.57}$$

$$\sum_{j \in J_s} x_{sj} - r_s \geq -M(1-v_s) \qquad \forall s \in S \tag{4.58}$$

$$r_s \leq M v_s \qquad \forall s \in S \tag{4.59}$$

$$r_s \geq 0 \qquad \forall s \in S \tag{4.60}$$

$$x_{sj} \in \{0,1\} \qquad \forall s \in S, \forall j \in J_s \tag{4.61}$$

$$v_s \in \{0,1\} \qquad \forall s \in S \tag{4.62}$$

$$u_{is} \in \{0,1\} \qquad \forall i \in I, \forall s \in S \tag{4.63}$$

$$w_{sji} \in \{0,1\} \qquad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.64}$$

$$y_{is} \in \{0,\ldots,|J_s|\} \qquad \forall i \in I, \forall s \in S \tag{4.65}$$

$$r_s \in \{0,\ldots,|J_s|\} \qquad \forall s \in S \tag{4.66}$$

$$z_s \in \{0,\ldots,V_s m_s\} \qquad \forall s \in S \tag{4.67}$$

$$z' \in \{0,\ldots,G \sum_{s \in S} m_S\} \tag{4.68}$$

Equation (4.44) represents the objective function. Constraints (4.45) again denote the set covering constraints. Constraints (4.46) guarantee that the V_s^L norm is satisfied for all depots and for those depots for which v_s is equal to 1, the V_s^H norm is satisfied. Constraints (4.47) make sure that the depots for which $v_s = 1$ holds, together host at least 90% of the crew. Constraints (4.48) ensure that the average number of unique kilometers per train driver over the entire country is at least G over the weekly planning horizon, or penalty costs are charged. Constraints (4.49) are needed to linearize the minimization part. The remaining constraints are required to assign the right values to the auxiliary variables (Constraints (4.50) to (4.60)) and to restrict the domains of all variables (Constraints (4.61) to (4.68)).

4.2 Adjusted Formulations

Since both formulations include many variables and difficult constraints, obtaining a solution to the crew scheduling problem with variation restrictions could be difficult. To avoid this risk, complicatedness of the problem is reduced by making several simplifying assumptions. These will be discussed below, including their impact on both formulations.

4.2.1 Simplifying Assumptions

One of the difficulties in solving the set covering problem with variation norms is that the number of drivers for the different depots is not known in advance, but depends on which duties are selected. This makes it difficult to solve the Lagrangian subproblem, since there is no clear way to relax constraints and rewrite the objective function such that the optimal solution is straightforward. Therefore, we decided to use a fixed number of drivers for every depot, denoted by m_s. This number is updated iteratively, as will be explained in Section 5.2.4.

Furthermore, the problem is complicated by the fact that there exist two norm values for variation that could be applied for a depot, namely a low norm value and a high norm value. Besides selecting duties such that the variation norms for the different depots are satisfied, depots should be assigned either the high norm value or the low norm value, such that at most 10 percent of the crew members are based at a depot with low norm value. Since this assignment makes solving the problem more difficult, we decided to make a simplifying assumption that for every depot it is known beforehand which variation norm value should be met, denoted by V_s. This assumption is reasonable, since it can either be deducted from geographical location of the depots or considered as general knowledge of planners for which depot a lower value should be used.

4.2.2 Method 1: Linearize Floor Function

When we adjust the first formulation using these assumptions, the following simplified version of the crew scheduling problem is obtained. Since certain multiplications of variables are eliminated, fewer auxiliary variables and constraints are needed. The remaining constraints serve the same purpose as in the original formulation.

$$\min \quad \sum_{s\in S}\left(\sum_{j\in J_s} c_{sj}x_{sj} + Pz_s\right) + Pz' \tag{4.69}$$

$$\text{s.t.} \quad \sum_{s\in S}\sum_{j\in J_s} a_{sjt}x_{sj} \geq 1 \qquad \forall t \in T \tag{4.70}$$

$$\sum_{i\in I} k_i\left(m_s\left(1-u_{is}\right)+25w_{is}\right) \geq V_s m_s - z_s \qquad \forall s \in S \tag{4.71}$$

$$0 \leq \sum_{j \in J_s} x_{sj} h_{sji} - 5q_{is} \leq 4 \quad \forall i \in I, \forall s \in S \tag{4.72}$$

$$\sum_{s \in S} \sum_{i \in I} k_i \left(m_s \left(1 - u_{is}\right) + 25 w_{is}\right) \geq G \sum_{s \in S} m_s - z' \tag{4.73}$$

$$m_s - 25 q_{is} \leq M u_{is} \quad \forall i \in I, \forall s \in S \tag{4.74}$$

$$-M\left(1 - u_{is}\right) \leq q_{is} - w_{is} \leq M\left(1 - u_{is}\right) \quad \forall i \in I, \forall s \in S \tag{4.75}$$

$$0 \leq w_{is} \leq M u_{is} \quad \forall i \in I, \forall s \in S \tag{4.76}$$

$$x_{sj} \in \{0, 1\} \quad \forall s \in S, \forall j \in J_s \tag{4.77}$$

$$q_{is} \in \{0, \ldots, q_{is}^{max}\} \quad \forall i \in I, \forall s \in S \tag{4.78}$$

$$u_{is} \in \{0, 1\} \quad \forall i \in I, \forall s \in S \tag{4.79}$$

$$w_{is} \in \{0, \ldots, q_{is}^{max}\} \quad \forall i \in I, \forall s \in S \tag{4.80}$$

$$z_s \in \{0, \ldots, V_s m_s\} \quad \forall s \in S \tag{4.81}$$

$$z' \in \{0, \ldots, G \sum_{s \in S} m_s\} \tag{4.82}$$

4.2.3 Method 2: Avoid Floor Function

Similar to the first formulation, the second formulation can be simplified using the aforementioned assumptions. Again, part of the auxiliary variables and constraints can now be disregarded.

$$\min \quad \sum_{s \in S} \left(\sum_{j \in J_s} c_{sj} x_{sj} + P z_s \right) + P z' \tag{4.83}$$

$$\text{s.t.} \quad \sum_{s \in S} \sum_{j \in J_s} a_{sjt} x_{sj} \geq 1 \quad \forall t \in T \tag{4.84}$$

$$\sum_{i \in I} k_i \left(m_s \left(1 - u_{is}\right) + 25 \left(\frac{\sum_{j \in J_s} w_{sji} h_{sji}}{5} - u_{is} \right) \right) \geq V_s m_s - z_s \quad \forall s \in S \tag{4.85}$$

$$\sum_{s \in S} \sum_{i \in I} k_i \left(m_s \left(1 - u_{is}\right) + 25 \left(\frac{\sum_{j \in J_s} w_{sji} h_{sji}}{5} - u_{is} \right) \right) \geq G \sum_{s \in S} m_s - z' \tag{4.86}$$

$$m_s - 25 \left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1 \right) \leq M u_{is} \quad \forall i \in I, \forall s \in S \tag{4.87}$$

$$w_{sji} \leq x_{sj} \quad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.88}$$

$$w_{sji} \leq u_{is} \quad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.89}$$

$$w_{sji} \geq x_{sj} + u_{is} - 1 \quad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.90}$$

$$x_{sj} \in \{0, 1\} \quad \forall s \in S, \forall j \in J_s \tag{4.91}$$

$$u_{is} \in \{0, 1\} \quad \forall i \in I, \forall s \in S \tag{4.92}$$

$$w_{sji} \in \{0, 1\} \quad \forall i \in I, \forall s \in S, \forall j \in J_s \tag{4.93}$$

$$z_s \in \{0, \ldots, V_s m_s\} \quad \forall s \in S \tag{4.94}$$

$$z' \in \{0, \ldots, G \sum_{s \in S} m_s\} \tag{4.95}$$

5 Solution Approach

In this section, we will describe the methods applied for solving the set covering problem with additional variation norms. First, we will provide a more detailed description of the solution approach for the set covering problem without variation constraints. Next, we will propose methods to solve the problem with variation restrictions by extending this solution approach. The algorithms described in this section are implemented in Java SE 8 (Oracle Corporation, Redwood City, CA, USA) and run on a quad-core DELL Precision T5810 with Intel Xeon E5 3.7 GHz processor and 16 GB RAM (Dell Inc., Round Rock, TX, USA).

5.1 Solving the Basic Set Covering Formulation

We will first give a brief description of the solution approach presented by Caprara et al. (1999) for the set covering formulation of the basic crew scheduling problem. In Sections 5.2 and 5.3 we will explain how this method is adapted to solve the crew scheduling problem in which variation constraints are included.

5.1.1 General Overview

Consider the following basic set covering problem. Note that notation is slightly different from our formulation in Section 2.2. Let N be the set of columns, $N = \{1, \dots, n\}$, and M the set of rows, $M = \{1, \dots, m\}$. The costs of selecting column $j \in N$ are given by c_j. Column $j \in N$ covers row $i \in M$ if $a_{ij} = 1$. The problem is formulated mathematically as follows:

$$\min \sum_{j \in N} c_j x_j \tag{5.1}$$

$$\text{s.t.} \sum_{j \in N} a_{ij} x_j \geq 1 \qquad i \in M \tag{5.2}$$

$$x_j \in \{0,1\} \quad j \in N \tag{5.3}$$

Furthermore, let J_i be the set of all columns covering row i and I_j the set of rows covered by column j. Hence, $J_i = \{j \in N : a_{ij} = 1\}$ and $I_j = \{i \in M : a_{ij} = 1\}$.

A heuristic, based on Lagrangian relaxation and a greedy algorithm, is applied to solve the set covering problem. This method iteratively applies a procedure consisting of three phases. In the first phase, a near-optimal vector of Lagrangian multipliers is found. In the second phase, a greedy heuristic is applied that selects duties based on this vector of near-optimal multipliers to obtain a feasible solution. Next, columns that are likely to be in an optimal solution are fixed. After this procedure is executed, a refining method is applied and the resulting problem is re-optimized. The algorithm continues while the objective value corresponding to the best solution found so far exceeds a certain paramater B times the best lower bound. The value of B is set to 1.0.

5.1.2 3PHASE Procedure

Subgradient Phase The aim of the *Subgradient phase* is to find a vector of near-optimal Lagrangian multipliers. That is, we want to find multipliers λ such that the lower bound on the objective value is maximized. Therefore, the following Lagrangian subproblem is solved for different values of the multipliers. In this subproblem, the set covering constraints are relaxed and incorporated in the objective function, such that only binary restrictions on variables x_j remain. This leads to the following formulation, where $c_j(\lambda) = c_j - \sum_{i \in I_j} \lambda_i a_{ij}$.

$$\min L(\lambda) = \sum_{j \in N} c_j(\lambda) x_j + \sum_{i \in M} \lambda_i \tag{5.4}$$

$$\text{s.t.} \quad x_j \in \{0,1\} \quad j \in N \tag{5.5}$$

An optimal solution to this problem is selecting all duties for which $c_j(\lambda)$ is negative. That is, $x_j(\lambda) = 1$ if $c_j(\lambda) < 0$, $x_j(\lambda) = 0$ if $c_j(\lambda) > 0$ and $x_j(\lambda) \in \{0,1\}$ if $c_j(\lambda) = 0$.

In the first iteration of the 3PHASE procedure, initial multipliers are obtained by taking for every task the minimum over all duties that cover this task of the division of the costs of the duty by the number of tasks covered by it (see Equation (5.6)).

$$\lambda_i^0 = \min_{j: a_{ij}=1} \frac{c_j}{|I_j|} \tag{5.6}$$

In the following iterations of the 3PHASE procedure, the best multiplier vector found in the previous iterations, denoted by λ^*, is used and slightly modified. First, multipliers corresponding to tasks covered by the duties that are fixed are removed, after which the remaining entries are adjusted.

$$\lambda_i^0 = (1 + \delta_i)\lambda_i^*, \text{ where } \delta_i \in [-0.1, 0.1] \tag{5.7}$$

During an iteration of the subgradient phase, initial multipliers are updated using Equation (5.8).

$$\lambda_i^{k+1} = \max\left\{\lambda_i^k + \alpha \frac{UB - L(\lambda^k)}{||s(\lambda^k)||^2} s_i(\lambda^k), 0\right\} \tag{5.8}$$

The upper bound (UB) is taken to be the currently best solution value of the set covering problem. The lower bound, $L(\lambda^k)$ is the solution value of the Lagrangian relaxation with multipliers λ^k. In Equation (5.8), $s(\lambda^k)$ refers to the subgradient vector corresponding to multipliers λ^k. For given λ^k, the subgradient for task i is calculated to be

$$s_i(\lambda^k) = 1 - \sum_{j \in J_i} x_j(\lambda^k) \tag{5.9}$$

where $x_j(\lambda^k)$ is the decision variable corresponding to $j \in J$ in the optimal solution of the Lagrangian problem associated with λ^k. Multiplier α has initial value 0.1. Every $p = 20$ iterations, the best and worst lower bound obtained during the last p iterations are compared. If the difference is more than 1% the value of α is halved, otherwise α is multiplied by a factor 1.5. The *Subgradient phase* terminates when the improvement of the lower bound in the last 300 iterations is smaller than 0.1%.

Heuristic Phase In the *Heuristic phase*, we search for feasible solutions based on the output of the *Subgradient phase*. A neighborhood of near-optimal multipliers is considered, for each of which a solution to the set covering problem is produced using a greedy heuristic (see Algorithm 2). In this heuristic duties with minimum score are selected until all tasks are covered by the selected duties.

Since the duty with minimum score is selected, favorable duties should be assigned lower scores than less favorable ones. This score is computed based on the Lagrangian costs of a duty and the number of uncovered tasks covered by it. Let μ_j be the number of uncovered tasks

Algorithm 2 Greedy

Initialization: $M^* := M$ is the set of currently uncovered tasks, $S := \varnothing$ is the set of currently selected duties
while $M^* \neq \varnothing$ **do**
  Compute score σ_j for each $j \in N \setminus S$
  Let $j^* \in N \setminus S$ be the column with minimum score
  $S := S \cup \{j^*\}$ and $M^* := M^* \setminus I_{j^*}$
end while

covered by duty $j \in N$ and $\gamma_j = c_j - \sum_{i \in I_j \cap M^*} \lambda_i^k$. The score for duty j is calculated using the following rule:

$$\sigma_j = \frac{\gamma_j}{\mu_j} \text{ if } \gamma_j > 0, \ \sigma_j = \gamma_j \mu_j \text{ if } \gamma_j < 0$$

To avoid recalculating γ_j and μ_j in every step, they can be updated after a duty has been selected. For every task $i \in I_{j^*} \cap M^*$ and for every duty $j \in J_i$ update $\gamma_j = \gamma_j + \lambda_i$ and $\mu_j = \mu_j - 1$.

The solution obtained by the greedy algorithm might contain redundant columns. Therefore, for every selected duty it is tested whether the solution remains feasible when this duty is deleted. The redundant column with highest cost is deleted and it is checked whether redundant columns are still present. This procedure continues until removing any duty makes the solution infeasible.

Column Fixing Phase In the final phase of the 3PHASE procedure, a subset of columns is selected to be fixed. Duties that have low costs and cover many tasks are likely to be selected in the final solution. Fixing these columns reduces the problem size, since the tasks covered by these duties do not need to be covered anymore. Let Q be the set of columns for which $c_j(\lambda^*) < -0.001$. Fix all columns $j \in Q$ for which there is a task that is only covered by column j and not by any of the other columns in Q. Next, apply the greedy heuristic and fix the first $\max\left\{\left\lfloor \frac{m}{200} \right\rfloor, 1\right\}$ columns that are selected.

5.1.3 Refining Procedure

For each column selected in the solution produced by the 3PHASE procedure, an estimate of its contribution to the overall gap between lower and upper bound is computed. The gap between the upper and lower bounds is given by Equation (5.10), where D denotes the set of selected duties.

$$\begin{aligned} GAP &= \sum_{j \in D} c_j - \left(\sum_{i \in M} \lambda_i^* + \sum_{j \in N : c_j(\lambda^*)} c_j(\lambda^*) \right) \\ &= \sum_{j \in D} \left(c_j(\lambda^*) + \sum_{i \in I_j} \lambda_i^* \right) - \sum_{i \in M} \lambda_i^* - \sum_{j \in N : c_j(\lambda^*)} c_j(\lambda^*) \\ &= \sum_{j \in D : c_j(\lambda^*) > 0} c_j(\lambda^*) + \sum_{j \in N \setminus D : c_j(\lambda^*) < 0} |c_j(\lambda^*)| + \sum_{i \in M} \lambda_i^* (|D \cap J_i| - 1) \end{aligned} \quad (5.10)$$

From this, we can compute an estimate of the contribution to the gap for every duty $j \in D$, denoted by δ_j (see Equation (5.11)).

$$\delta_j = \max\{c_j(\lambda^*), 0\} + \sum_{i \in I_j} \lambda_i^* \frac{|S \cap J_i| - 1}{|S \cap J_i|} \tag{5.11}$$

Columns are sorted based on increasing value of δ. Duties $j_1, j_2, \ldots$ are added until $\frac{|\cup_{j=j_1}^{j^*} I_j|}{|T|} \geq \pi$. The value of π determines the percentage of rows removed after the columns have been fixed. The selected columns are fixed and the 3PHASE procedure is again applied.

5.2 Extending the Crew Scheduling Problem with Variation Norms per Depot

In this section, we will present the different methods for solving the crew scheduling problem with variation restrictions per depot. First, we describe the basic model, which is used as a reference for the two newly developed methods. Next, we describe how the method discussed in Section 5.1 is adjusted to incorporate variation norms. First, since new constraints are added to the problem, the Lagrangian subproblem changes. Second, in the heuristic phase, the greedy procedure is adapted, because we want to find solutions that do not only satisfy the set covering constraints, but also the variation norms. Since the changes to the model are different for the two previously described formulations, we will go into detail about both approaches. First, we will discuss how the method is adjusted when we use the formulation in which the floor function is linearized exactly. After that, we will describe the changes to the basic model for the formulation in which $\lfloor x \rfloor$ is replaced by $x - 1$.

5.2.1 Method 0: Basic Model

To be able to compare the performance of the two developed methods, we will use the method described in Section 5.1 as a reference. In this solution approach no variation restrictions are considered. However, to quantify violations of the restrictions, we calculate penalty costs.

5.2.2 Method 1: Linearize Floor Function

We will now consider the changes to the different parts of the algorithm when we use the formulation in which the floor function is linearized exactly.

Lagrangian Subproblem When we add the variation restrictions to the original crew scheduling problem, the Lagrangian subproblem is given by Equations (5.12) to (5.18), where Equation (5.13) represents the variation restriction.

$$\min\ L(\lambda) = \sum_{s \in S} \left(P z_s + \sum_{j \in J_s} c_{sj}(\lambda) x_{sj} \right) + \sum_{t \in T} \lambda_t \left(\sum_{s \in S} \sum_{j \in J_s} a_{sjt} x_{sj} - 1 \right) \tag{5.12}$$

$$\text{s.t.} \quad \sum_{i \in I} k_i \min\{m_s, q_{is}\} \geq V_s m_s - z_s \tag{5.13}$$

$$\sum_{j \in J_s} h_{sji} x_{sj} - 5q_{is} \geq 0 \qquad \forall\, s \in S, i \in I \tag{5.14}$$

$$\sum_{j \in J_s} h_{sji} x_{sj} - 5q_{is} \leq 4 \qquad \forall\, s \in S, i \in I \tag{5.15}$$

$$x_{sj} \in \{0, 1\} \qquad j \in N \tag{5.16}$$

$$q_{is} \in \{0, \ldots, q_{is}^{max}\} \quad s \in S,\ i \in I \tag{5.17}$$

$$z_s \in \{0, \ldots, V_s m_s\} \quad s \in S \tag{5.18}$$

Since the optimal solution to the subproblem is not straightforward, the constraints concerning variation norms are relaxed. Corresponding Lagrangian multipliers θ_s are introduced. The objective function of the Lagrange subproblem can be written as follows.

$$\min \sum_{s \in S} \left(P z_s + \sum_{j \in J_s} c_{sj}(\lambda) x_{sj} \right) - \sum_{t \in T} \lambda_t \left(\sum_{s \in S} \sum_{j \in J_s} a_{sjt} x_{sj} - 1 \right)$$
$$- \sum_{s \in S} \theta_s \left(\sum_{i \in I} k_i \min\{m_s, q_{is}\} - (V_s m_s - z_s) \right) \tag{5.19}$$

However, since the variation constraint contains a minimization function, the objective function needs to be linearized. Instead of adding the additional constraints described in Section 4.1.2, we can apply some modeling tricks. When turning the objective function into a maximization problem, we obtain a maximization over a minimization. In this case, the minimization part can be replaced by an auxiliary variable, y_{is}, that is smaller than or equal to both of the terms in the minimization function. Since the aim is to maximize the entire objective function, hence maximize this term, no additional constraints are needed. This leads to the following problem formulation.

$$\max\ L(\lambda) = - \sum_{s \in S} \left(P z_s + \sum_{j \in J_s} c_{sj}(\lambda) x_{sj} \right) + \sum_{t \in T} \lambda_t \left(\sum_{s \in S} \sum_{j \in J_s} a_{sjt} x_{sj} - 1 \right)$$
$$+ \sum_{s \in S} \theta_s \left(\sum_{i \in I} k_i y_{is} - (V_s m_s - z_s) \right) \tag{5.20}$$

$$\text{s.t.} \quad \sum_{j \in J_s} h_{sji} x_{sj} - 5q_{is} \geq 0 \qquad \forall\, s \in S, i \in I \tag{5.21}$$

$$\sum_{j \in J_s} h_{sji} x_{sj} - 5q_{is} \leq 4 \qquad \forall\, s \in S, i \in I \tag{5.22}$$

$$y_{is} \leq m_s \qquad \forall\, s \in S, i \in I \tag{5.23}$$

$$y_{is} \leq 25 q_{is} \qquad \forall\, s \in S, i \in I \tag{5.24}$$

$$x_{sj} \in \{0, 1\} \qquad j \in N \tag{5.25}$$

$$q_{is} \in \{0, \ldots, q_{is}^{max}\} \quad s \in S,\ i \in I \tag{5.26}$$

$$y_{is} \in \{0, \ldots, m_s\} \quad s \in S,\ i \in I \tag{5.27}$$

$$z_s \in \{0, \ldots, V_s m_s\} \quad s \in S \tag{5.28}$$

Since Constraints (5.21) and (5.22) contain variables x_{sj} and q_{is}, the problem remains difficult to solve. In addition, since the values of variables q_{is} are not fixed, Constraints (5.24) are not trivial. Therefore, these constraint sets are relaxed to be able to assign appropriate values to the variables. Lagrange multipliers κ_{is} are introduced for Constraints (5.21) and ξ_{is} for Constraints (5.22). Finally, multipliers ζ_{is} are introduced when relaxing Constraints (5.24). After rewriting the objective function, we obtain the following subproblem formulation.

$$\max\ L(\lambda) = \sum_{s\in S}\left(\sum_{j\in J_s}\left[\sum_{t\in T}\lambda_t a_{sjt} + \sum_{i\in I}\left(\kappa_{is}-\xi_{is}\right)h_{sji} - c_{sj}\right]x_{sj} + (\theta_s - P)z_s + \sum_{i\in I}\left[\left(\theta_s k_i - \zeta_{is}\right)y_{is} + \left(25\zeta_{is} + 5\left(\xi_{is}-\kappa_{is}\right)\right)q_{is} + 4\xi_{is}\right] - \theta_s V_s m_s\right) - \sum_{t\in T}\lambda_t \tag{5.29}$$

$$\text{s.t.} \quad y_{is} \le m_s \quad \forall\, s\in S, i\in I \tag{5.30}$$
$$x_{sj}\in\{0,1\} \quad j\in N \tag{5.31}$$
$$q_{is}\in\{0,\dots,q_{is}^{max}\} \quad s\in S,\ i\in I \tag{5.32}$$
$$y_{is}\in\{0,\dots,m_s\} \quad s\in S,\ i\in I \tag{5.33}$$
$$z_s\in\{0,\dots,V_s m_s\} \quad s\in S \tag{5.34}$$

For this problem, finding an optimal solution is straightforward. Since we are dealing with a maximization problem, we want to assign values to variables such that their values are as large as possible when the coefficient is positive and as small as possible when it is negative. This leads to the following assignments:

$$x_{sj} = \begin{cases} 1 & \text{if} \quad \sum_{t\in T}\lambda_t a_{sjt} + \sum_{i\in I}\left(\kappa_{is}-\xi_{is}\right)h_{sji} - c_{sj} > 0 \\ 0 & \text{if} \quad \sum_{t\in T}\lambda_t a_{sjt} + \sum_{i\in I}\left(\kappa_{is}-\xi_{is}\right)h_{sji} - c_{sj} \le 0 \end{cases}$$

Using similar reasoning, the values of y_{is} and z_s in an optimal solution to the Lagrangian subproblem can be obtained as follows:

$$y_{is} = \begin{cases} m_s & \text{if} \quad \theta_s k_i - \zeta_{is} > 0 \\ 0 & \text{if} \quad \theta_s k_i - \zeta_{is} \le 0 \end{cases}$$

$$z_s = \begin{cases} V_s m_s & \text{if} \quad \theta_s - P > 0 \\ 0 & \text{if} \quad \theta_s - P \le 0 \end{cases}$$

The values of variables q_{is} can be determined in the same way. When its coefficient is positive, q_{is} is assigned its maximum value q_{is}^{max}.

$$q_{is} = \begin{cases} q_{is}^{max} & \text{if} \quad 25\zeta_{is} + 5\left(\xi_{is}-\kappa_{is}\right) > 0 \\ 0 & \text{if} \quad 25\zeta_{is} + 5\left(\xi_{is}-\kappa_{is}\right) \le 0 \end{cases}$$

Update Multipliers The Lagrangian multipliers are updated similar to those corresponding to the set covering constraints in the solution approach of the basic model. In the first iteration of the 3PHASE procedure, the multipliers are initialized at 0. In further iterations, their initial value is equal to the optimal multipliers of the previous iteration that are slightly altered (Equation (5.35) to (5.38)). The optimal values are given by θ^*, ζ^*, κ^* and ξ^*.

$$\theta_s = (1+\delta_s)\theta_s^*, \text{ where } \delta_s \in [-0.1, 0.1] \tag{5.35}$$

$$\zeta_{is} = (1 + \delta_{is})\zeta^*_{is}, \text{ where } \delta_{is} \in [-0.1, 0.1] \tag{5.36}$$

$$\kappa_{is} = (1 + \delta_{is})\kappa^*_{is}, \text{ where } \delta_{is} \in [-0.1, 0.1] \tag{5.37}$$

$$\xi_{is} = (1 + \delta_{is})\xi^*_{is}, \text{ where } \delta_{is} \in [-0.1, 0.1] \tag{5.38}$$

During an iteration of the 3PHASE procedure, the multipliers are updated using the same approach as is used for λ. This requires the computation of the subgradient vectors corresponding to the multipliers (see Equations (5.39) to (5.42)).

$$s^{\theta}_s(\theta^k, \zeta^k) = V_s m_s - z_s - \sum_{i \in I} k_i y_{is} \tag{5.39}$$

$$s^{\zeta}_{is}(\theta^k, \zeta^k, \kappa^k, \xi^k) = y_{is} - 25q_{is} \tag{5.40}$$

$$s^{\kappa}_{is}(\lambda^k, \zeta^k, \kappa^k, \xi^k) = 5q_{is} - \sum_{j \in J_s} h_{sji} x_{sj} \tag{5.41}$$

$$s^{\xi}_{is}(\lambda^k, \zeta^k, \kappa^k, \xi^k) = \sum_{j \in J_s} h_{sji} x_{sj} - 5q_{is} - 4 \tag{5.42}$$

The updating formulas are given by Equations (5.43) till (5.46), where $L(\lambda^k, \theta^k, \zeta^k, \kappa^k, \xi^k)$ represents the lower bound for the given multipliers.

$$\theta^{k+1}_s = \min\left\{\theta^k_s + \alpha \frac{UB - L(\lambda^k, \theta^k, \zeta^k, \kappa^k, \xi^k)}{||s^{\theta}_s(\theta^k, \zeta^k)||^2} s^{\theta}_s(\theta^k, \zeta^k), 0\right\} \tag{5.43}$$

$$\zeta^{k+1}_{is} = \min\left\{\zeta^k_{is} + \alpha \frac{UB - L(\lambda^k, \theta^k, \zeta^k, \kappa^k, \xi^k)}{||s^{\zeta}_{is}(\theta^k, \zeta^k, \kappa^k, \xi^k)||^2} s^{\zeta}_{is}(\theta^k, \zeta^k, \kappa^k, \xi^k), 0\right\} \tag{5.44}$$

$$\kappa^{k+1}_{is} = \min\left\{\kappa^k_{is} + \alpha \frac{UB - L(\lambda^k, \theta^k, \zeta^k, \kappa^k, \xi^k)}{||s^{\kappa}_{is}(\lambda^k, \zeta^k, \kappa^k, \xi^k)||^2} s^{\kappa}_{is}(\lambda^k, \zeta^k, \kappa^k, \xi^k), 0\right\} \tag{5.45}$$

$$\xi^{k+1}_{is} = \min\left\{\xi^k_{is} + \alpha \frac{UB - L(\lambda^k, \theta^k, \zeta^k, \kappa^k, \xi^k)}{||s^{\xi}_{is}(\lambda^k, \zeta^k, \kappa^k, \xi^k)||^2} s^{\xi}_{is}(\lambda^k, \zeta^k, \kappa^k, \xi^k), 0\right\} \tag{5.46}$$

Greedy Heuristic Several changes need to be made to the greedy heuristic to obtain a good solution to the problem. First, the stopping criteria of this procedure needs to be adjusted. In the original greedy heuristic, duties are added to the set of selected duties until all tasks have been covered. When we add variation constraints to the problem, we need to continue adding duties until both constraint sets are satisfied. That is, until all tasks are covered by the selected duties and the variation constraints for the different depots are satisfied. In case it is not possible to satisfy the variation constraints for all depots with the available duties, penalty costs will be present in the final solution. If this is the case, we continue adding duties as long as costs were reduced in previous iterations.

Besides adjusting the stopping criteria, we also want to change the computation of scores for the different duties. In the original method, scores are calculated such that duties with low costs covering a large number of uncovered tasks are selected. Now we adjust this computation in such a way that, besides favoring duties that cover uncovered tasks, we consider how much variation a duty can add to the corresponding depot. Several score calculations are considered. Since we select a duty with minimum score, we would like the score of a duty to be low when the variation restriction for the corresponding depot is not satisfied and this duty can add to the variation. Therefore, the following score computation methods are tested. Computation method (5.47) focuses on the effect on variation, while (5.48) focuses more on uncovered tasks that are covered by the considered duty. Also, computation method (5.49) pays more attention to variation and uncovered tasks than to costs of the duty.

$$\sigma_{sj} = \frac{c_{sj}}{\mu_{sj}} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s \tag{5.47}$$

$$\sigma_{sj} = \begin{cases} \frac{c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s}{\mu_{sj}} & \text{if } c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s > 0 \\ \left(c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s\right)\mu_{sj} & \text{if } c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s \leq 0 \end{cases} \tag{5.48}$$

$$\sigma_{sj} = \frac{c_{sj}}{\mu_{sj} + \sum_{t \in T_j \cap M^*} \lambda_t + \beta\theta_s} \tag{5.49}$$

It is possible that all tasks are covered, hence $\mu_{sj} = 0$ for all duties, but some variation restrictions are not satisfied. In Equations (5.47) and (5.48) this hinders the score calculation. In such cases, if the variation restriction is not met for at least one depot, we add a duty for one of these depots, if there are duties remaining. If violations are present for certain depots but no more duties are available, penalty costs are calculated based on the variation present in the current schedule. Since the solution may contain duties that neither add to the variation, nor cover otherwise uncovered tasks, it is checked whether duties can be removed from the solution without increasing costs.

5.2.3 Method 2: Approximate Floor Function

We continue by describing how the approach described in Section 5.1 is adjusted based on the second formulation of the problem. First, we will explain how the Lagrangian subproblem is altered, after which we describe updating of the multipliers. Finally, we will discuss changes to the greedy heuristic.

Lagrangian Subproblem Similar to Section 5.2.2, we let m_s and V_s denote the number of drivers for depot s and the norm value for depot s, respectively. Furthermore, we introduce variable $y_{is} = \min\left\{m_s, 25\left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1\right)\right\}$. When the variation constraints are added to the Lagrangian subproblem of the basic model, the following formulation is obtained.

$$\min\ L(\lambda) = \sum_{s \in S}\left(Pz_s + \sum_{j \in J_s} c_{sj}(\lambda)x_{sj}\right) + \sum_{t \in T} \lambda_t \tag{5.50}$$

$$s.t. \sum_{i \in I} k_i y_{is} \geq V_s m_s - z_s \qquad s \in S \tag{5.51}$$

$$y_{is} = \min\left\{m_s, 25\left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1\right)\right\} \qquad s \in S,\ i \in I \tag{5.52}$$

$$x_{sj} \in \{0, 1\} \qquad s \in S,\ j \in J_s \tag{5.53}$$

$$y_{is} \in \{0, \ldots, m_s\} \qquad s \in S,\ i \in I \tag{5.54}$$

$$z_s \in \{0, \ldots, V_s m_s\} \qquad s \in S \tag{5.55}$$

To be able to solve the Lagrangian subproblem, variation constraints (5.51) are relaxed and corresponding parameters θ_s are introduced. However, we need to take into account that y_{is} represents the nonlinear term $\min\left\{m_s, 25\left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1\right)\right\}$, which is dependent on variables

x_{sj}. To overcome this problem we apply the same technique as was used in Section 5.2.2, where we rewrite the subproblem to the corresponding maximization problem. The objective function is then given by Equation (5.56).

$$\max \sum_{s\in S}\left(\sum_{j\in J_s}\left(\sum_{t\in T} a_{sjt}\lambda_t - c_{sj}\right)x_{sj} + (\theta_s - P)z_s + \theta_s \sum_{i\in I} k_i y_{is} - \theta_s V_s m_s\right) - \sum_{t\in T}\lambda_t \tag{5.56}$$

Since maximizing y_{is} denotes maximizing over a minimum, we need Constraints (5.57) and (5.58) to make sure y_{is} is specified correctly. Since the objective is maximization and the coefficient of y_{is} is positive, it is not necessary to include constraints restricting y_{is} to be larger than or equal to the minimum of the two terms.

$$y_{is} \leq m_s \tag{5.57}$$

$$y_{is} \leq 25\left(\frac{\sum_{j\in J_s} x_{sj}h_{sji}}{5} - 1\right) \tag{5.58}$$

In contrast to the Lagrangian subproblem of the basic model, solving this problem is not straightforward. Therefore, Constraints (5.58) are relaxed and corresponding Lagrangian multipliers ψ_{is} are introduced. This results in the following objective function for the Lagrangian subproblem.

$$\max \sum_{s\in S}\left(\sum_{j\in J_s}\left(\sum_{t\in T}\lambda_t a_{sjt} - c_{sj} + 25\sum_{i\in I}\psi_{is}\frac{h_{sji}}{5}\right)x_{sj} + (\theta_s - P)z_s + \right.$$
$$\left.\sum_{i\in I}(\theta_s k_i - \psi_{is})\, y_{is} - \theta_s V_s m_s - 25\sum_{i\in I}\psi_{is}\right) - \sum_{t\in T}\lambda_t \tag{5.59}$$

Since the only remaining constraints are of the form $y_{is} \leq m_s$ and those restricting the domains of the variables, finding an optimal solution for the Lagrangian subproblem is straightforward. Since the problem is now a maximization problem, we want to choose x_{sj} as large as possible when its coefficient is positive and as small as possible when its coefficient is negative. Therefore, the following rules can be applied:

$$x_{sj} = \begin{cases} 1 & \text{if} \quad \sum_{t\in T}\lambda_t a_{sjt} - c_{sj} + 25\sum_{i\in I}\psi_{is}\frac{h_{sji}}{5} > 0 \\ 0 & \text{if} \quad \sum_{t\in T}\lambda_t a_{sjt} - c_{sj} + 25\sum_{i\in I}\psi_{is}\frac{h_{sji}}{5} \leq 0 \end{cases}$$

Using similar reasoning, the values of y_{is} and z_s in an optimal solution can be obtained as follows:

$$y_{is} = \begin{cases} m_s & \text{if} \quad \theta_s k_i - \psi_{is} > 0 \\ 0 & \text{if} \quad \theta_s k_i - \psi_{is} \leq 0 \end{cases}$$

$$z_s = \begin{cases} V_s m_s & \text{if} \quad \theta_s - P > 0 \\ 0 & \text{if} \quad \theta_s - P \leq 0 \end{cases}$$

Update Multipliers In the first iteration of the 3PHASE procedure, the multipliers are set equal to 0. In following iterations their initial value is obtained by slightly modifying the optimal multipliers of the previous iterations, which are given by θ^* and ψ^*, as described by Equations (5.60) and (5.61).

$$\theta_s = (1+\delta_s)\theta_s^*, \text{ where } \delta_s \in [-0.1, 0.1] \tag{5.60}$$

$$\psi_{is} = (1 + \delta_{is})\psi^*_{is}, \text{ where } \delta_{is} \in [-0.1, 0.1] \tag{5.61}$$

During the 3PHASE procedure the multipliers are updated using the same approach as for λ in the basic problem (see Equations (5.62) and (5.63)).

$$\theta_s^{k+1} = \min\left\{\theta_s^k + \alpha\frac{UB - L(\lambda^k, \theta^k, \psi^k)}{||s_s^\theta(\theta^k, \psi^k)||^2}s_s^\theta(\theta^k, \psi^k), 0\right\} \tag{5.62}$$

$$\psi_{is}^{k+1} = \min\left\{\psi_{is}^k + \alpha\frac{UB - L(\lambda^k, \theta^k, \psi^k)}{||s_{is}^\psi(\lambda^k, \theta^k, \psi^k)||^2}s_{is}^\psi(\lambda^k, \theta^k, \psi^k), 0\right\} \tag{5.63}$$

The term $s_s^\theta(\theta^k, \psi^k)$ refers to the element corresponding to depot s of the subgradient vector calculated using Lagrange multipliers vectors θ^k and ψ^k. Similarly, $s_{is}^\psi(\lambda^k, \theta^k, \psi^k)$ is the element of the subgradient vector calculated using λ^k, θ^k and ψ^k that corresponds to the combination of infrastructure piece i and depot s. These values are calculated as follows:

$$s_s^\theta(\theta^k, \psi^k) = V_s m_s - z_s - \sum_{i \in I} k_i y_{is} \tag{5.64}$$

$$s_{is}^\psi(\lambda^k, \theta^k, \psi^k) = y_{is} - \sum_{j \in J_s} h_{sji} x_{sj} \tag{5.65}$$

Greedy Heuristic The stopping criteria for the greedy heuristic are adjusted in the same way as described in Section 5.2.2 for Method 1. In addition, we adjust the method applied for calculating scores, such that violation of the variation restrictions is taken into account. There are again different ways to compute scores using these variables. The following scoring rules will again be considered:

$$\sigma_{sj} = \frac{c_{sj}}{\mu_{sj}} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s \tag{5.66}$$

$$\sigma_{sj} = \begin{cases} \frac{c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s}{\mu_{sj}} & \text{if } c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s > 0 \\ \left(c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s\right)\mu_{sj} & \text{if } c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s \leq 0 \end{cases} \tag{5.67}$$

$$\sigma_{sj} = \frac{c_{sj}}{\mu_{sj} + \sum_{t \in T_j \cap M^*} \lambda_t + \beta\theta_s} \tag{5.68}$$

The same procedure as described in Section 5.2.2 is applied when all tasks are satisfied, but variation norms are not met. In addition, we check whether duties can be removed from the solution.

5.2.4 Update Number of Drivers

In order to better capture the relation between the number of drivers per depot and the selected duties, the number of drivers is updated after the greedy heuristic has produced a solution. However, during each iteration of the greedy heuristic the objective value of the current solution is compared to the best known solution. In this comparison, it could be the case that the variation for the depots in this solution is over- or underestimated because the number of drivers did not relate correctly to the selected duties. To overcome this problem, the objective value corresponding to the solution generated by the greedy heuristic is recalculated using the number of drivers indicated by the selected duties. This allows for a fairer comparison between the solutions obtained in different iterations.

5.3 Solving the Crew Scheduling Problem with Variation Norms per Depot and Average Norm

In this section, we extend the preceding problem by including an average variation norm. To measure average variation, we take an average over the individual variation measures of the different depots weighted by their number of drivers. Let G be the norm value for the average variation constraint. First, we again discuss the basic model, next we describe both developed methods.

5.3.1 Method 0: Basic Model

We will again use the method described in Section 5.1 as a reference to compare performance of the two new methods. In this solution approach no variation restrictions are considered, but we calculate penalty costs for violations both of the variation restrictions per depot and of the average variation restriction.

5.3.2 Method 1: Linearize Floor Function

We will first describe how the crew scheduling algorithm is adapted when we add the average variation norm to the method based on the first formulation. In this section we will describe how the Lagrangian subproblem is adjusted and which changes to the greedy heuristic are needed.

Lagrangian Subproblem To incorporate the average variation norm, we add the restriction given by Equation (5.69) to the Lagrangian subproblem that was given by Equations (5.12) to (5.18) in Section 5.2.2.

$$\sum_{s \in S} \sum_{i \in I} k_i y_{is} \geq G \sum_{s \in S} m_s - z' \tag{5.69}$$

This constraint contains variable y_{is}, which represents the minimum of m_s and q_{is}, and is therefore a difficult constraint. Hence, the Lagrangian subproblem is solved by first relaxing this constraint and introducing corresponding Lagrangian multiplier η. This leads to the following subproblem:

$$\max\ L(\lambda) = \sum_{s \in S} \left(\sum_{j \in J_s} \left[\sum_{t \in T} \lambda_t a_{sjt} + \sum_{i \in I} (\kappa_{is} - \xi_{is})\, h_{sji} - c_{sj} \right] x_{sj} + (\theta_s - P) z_s - \eta G m_s + \right.$$
$$\left. \sum_{i \in I} \left[\left((\theta_s + \eta)\, k_i - \zeta_{is} \right) y_{is} + \left(25\zeta_{is} + 5\left(\xi_{is} - \kappa_{is}\right) \right) q_{is} + 4\xi_{is} \right] - \theta_s V_s m_s \right) + (\eta - P) z' - \sum_{t \in T} \lambda_t \tag{5.70}$$

$$\begin{aligned}
\text{s.t.} \quad && y_{is} \leq m_s && \forall\, s \in S, i \in I && (5.71)\\
&& x_{sj} \in \{0, 1\} && j \in N && (5.72)\\
&& q_{is} \in \{0, \dots, q_{is}^{max}\} && s \in S,\ i \in I && (5.73)\\
&& y_{is} \in \{0, \dots, m_s\} && s \in S,\ i \in I && (5.74)\\
&& z_s \in \{0, \dots, V_s m_s\} && s \in S && (5.75)\\
&& z' \in \{0, \dots, G \sum_{s \in S} m_s\} && && (5.76)
\end{aligned}$$

Since the coefficients of variables x_{sj}, q_{is} and z_s have not changed, their values are determined in the same way as for the problem without average variation norm. For the y_{is} variables, the coefficients have changed and therefore their values are determined as follows:

$$y_{is} = \begin{cases} m_s & \text{if} \quad (\theta_s + \eta)\, k_i - \zeta_{is} > 0 \\ 0 & \text{if} \quad (\theta_s + \eta)\, k_i - \zeta_{is} \leq 0 \end{cases}$$

In addition, variable z' is introduced in the subproblem. The value of this variable is determined as follows:

$$z' = \begin{cases} G \sum_{s \in S} m_s & \text{if} \quad \eta - P > 0 \\ 0 & \text{if} \quad \eta - P \leq 0 \end{cases}$$

Greedy Heuristic Similar to when the variation constraints per depot were added to the problem, the stopping criteria of the greedy heuristic need to be adapted. In addition to the previously mentioned conditions (task coverage and satisfying variation per depot), we need to continue adding duties until the average variation norm has been satisfied. Similar to the case in which only variation norms per depot are considered, penalty costs are added when it is not possible to satisfy all variation norms with the given duties.

The method for computing scores of duties is not adjusted. The reason for this decision is that duties that are favorable because they increase individual variation of the depots are also useful for increasing average variation over all depots.

5.3.3 Method 2: Avoid Floor Function

In this section we describe how different parts of the algorithm are adapted when we add the average variation norm to the solution method based on the second model formulation.

Lagrangian Subproblem We now add the average variation norm given by Equation (5.77) to the Lagrangian subproblem with objective function (5.59) and constraints $y_{is} \leq m_s$ for every $i \in I$ and $s \in S$.

$$\sum_{s \in S} \sum_{i \in I} k_i y_{is} \geq G \sum_{s \in S} m_s - z' \tag{5.77}$$

Since $y_{is} = \min \left\{ m_s, 25 \left(\frac{\sum_{j \in J_s} x_{sj} h_{sji}}{5} - 1 \right) \right\}$, this constraint makes the subproblem difficult to solve. Therefore, we introduce Lagrangian multiplier η to relax constraint $\sum_{s \in S} \sum_{i \in I} k_i y_{is} \geq G \sum_{s \in S} m_s - z'$. The following term is then added to the objective function of the Lagrangian subproblem:

$$-\eta \left(G \sum_{s \in S} m_s - z' - \sum_{s \in S} \sum_{i \in I} k_i y_{is} \right) \tag{5.78}$$

This results in objective function (5.79) to be maximized, subject to constraints $y_{is} \leq m_s$ and those restricting the domains of all variables.

$$\max \sum_{s \in S} \left(\sum_{j \in J_s} \left(\sum_{t \in T} \lambda_t a_{sjt} - c_{sj} + 25 \sum_{i \in I} \psi_{is} \frac{h_{sji}}{5} \right) x_{sj} + \sum_{i \in I} \left((\theta_s + \eta) k_i - \psi_{is} \right) y_{is} \right.$$
$$\left. + (\theta_s - P) z_s - \theta_s V_s m_s - \eta G m_s - 25 \sum_{i \in I} \psi_{is} \right) + (\eta - P) z' - \sum_{t \in T} \lambda_t \tag{5.79}$$

The coefficients of variables x_{sj} and z_s have not changed, hence their optimal value can be determined in the same way as for the problem without average variation restriction. For variables y_{is} the coefficient has changed, therefore the rule for determining their optimal values needs to be adjusted. The value of y_{is} is now given by:

$$y_{is} = \begin{cases} m_s & \text{if} \quad (\theta_s + \eta)k_i - \psi_{is} > 0 \\ 0 & \text{if} \quad (\theta_s + \eta)k_i - \psi_{is} \leq 0 \end{cases}$$

Since additional variable z' is now added to the problem, we need to specify how to determine its value. This can be done as follows.

$$z' = \begin{cases} G \sum_{s \in S} m_s & \text{if} \quad \eta - P > 0 \\ 0 & \text{if} \quad \eta - P \leq 0 \end{cases}$$

Greedy Heuristic The stopping criteria used for the greedy heuristic in case the average variation norm needs to be satisfied is adjusted in the same way as described in Section 5.3.2. Besides satisfying the set covering constraints and variation constraints per depot, the average variation norm should be met before the greedy procedure terminates, or penalty costs should be included when no feasible solution is found. In this case, we again continue until no decrease in costs is observed for several iterations.

For the same reason as mentioned in Section 5.3.2, the score calculation method is not adjusted compared to those described in Section 5.2.3.

6 Test Instances

Test instances are needed in order to evaluate performance of the methods. First of all, data on tasks that need to be covered is needed. For each task, information on its start and end locations and times is required. In addition, information is needed on which infrastructure pieces are included in a certain task. A piece of infrastructure is a part of the railway network between two stations with a corresponding length. To avoid having superfluous variables, the length of infrastructure pieces is maximized while making sure tasks either cover a piece entirely or not at all.

Two different types of datasets are used during this study. First, artificial datasets of varying sizes are used during the construction of the model. Second, performance of the model is evaluated on instances generated from real life data.

6.1 Artificial Instances

Artificial instances of different sizes were created to test the methods. For each instance, a randomly chosen number of depots and infrastructure pieces is available. We created instances with 50, 100, 200, 300 and 400 tasks. The set of duties available to choose from was created by randomly assigning tasks to duties. Each task must be covered by at least one of these duties, such that a feasible task cover can be found. For each duty a depot is selected. Next, we determined the number of tasks to be assigned to the duty, which is chosen uniformly on the interval [2,8]. We then randomly selected this number of tasks from the entire set of tasks. The costs of a duty were determined using a uniform distribution on the interval [500,600]. Details about the number of tasks and duties, average number of tasks per duty, the number of depots and infrastructure pieces and average length per piece of infrastructure can be found in Table 6.1. From now on, these datasets will be referred to using the number of tasks they contain. Note that instance 50 is somewhat different from the other instances.

Table 6.1: Characteristics artificial instances

Tasks	Duties	Avg tasks per duty	Depots	Infrapieces	Avg length infrapiece
50	151	1.96	3	10	4.60
100	151	4.57	5	40	2.83
200	301	4.62	8	70	3.53
300	501	4.47	15	80	2.96
400	1001	4.57	15	80	3.69

6.1.1 Variation Norm Values

To test the performance of the algorithm when variation norms are included in the model, we need to specify the required norm values. To determine appropriate values, we observe the level of variation present in the solution of the basic model for each dataset. Based on these values, we set the norm values for variation per depot and average variation, such that the variation constraints make the problem somewhat more restricted.

The minimum and maximum variation per depot and average variation for all depots are given in Table 6.2. In the last two columns the selected norm values are reported. These values are chosen such that the variation norm per depot is restrictive for at least part of the depots and the average variation norm is above average variation in the basic solution.

Table 6.2: Variation per depot for artificial instances without variation norm

Tasks	Avg var	Minimum var	Maximum var	Norm per depot	Avg norm
50	40.69	37.00	45.00	42	45
100	60.96	0	113.00	40	70
200	221.00	0	247.00	100	250
300	187.83	0	237.00	30	200
400	271.68	0	295.00	30	280

6.2 Real Life Instances

After testing the performance of the models on artificial instances of different sizes, we apply the models to two instances created based on real life data, one of which contains tasks for a single day and one for multiple days.

The first instance using real life data contains all tasks that are currently assigned to depot Utrecht on a certain weekday. These tasks are used as the set of tasks that need to be covered. The model is used to assign these tasks to two depots, namely Utrecht (Ut) and Amersfoort (Amf). Each task starts at a given location at a given time and ends at a different location at a given time. The set of start and end locations consists of all transfer stations, i.e. larger stations where a train driver can be relieved. The available set consists of 1442 tasks that need to be assigned to duties.

Based on these tasks, duties were created to select from, such that every task is covered by at least one duty. Since the focus of this study is on selecting duties from a set of available duties and not on creation of duties, a simple procedure was applied for constructing these duties. On average, 15 tasks are assigned to one duty. Duties were created by selecting a depot (Utrecht or Amersfoort) and a task leaving from this depot. Possible further tasks have as start location the destination of the previous task and start within 30 minutes after that task has ended. The final task must return to the depot. If such duty was identified, it was added to the set of duties, otherwise we started again with an empty duty. No other rules concerning individual duties or the entire schedule have been taken into account, since the focus of this study is on incorporating the variation restrictions. The cost of duties is expressed as their duration in minutes. This instance contains 2000 duties that can be selected.

The second instance using real life data contains tasks that need to be executed during three consecutive working days. For simplicity, it is assumed that the same tasks that were used for the instance with one working day need to be covered on each of the three days. The same set of duties is assumed to be available for all three working days, hence 6000 duties are available. This instance will be referred to as 'Multi-day instance'.

Next, the railway network covered by these tasks is divided into infrastructure pieces. For each piece of infrastructure the length and its start and end location are required. It is assumed that the trajectory covered by a task can be divided into infrastructure pieces by splitting the trajectory at intermediate stations, where there is a possibility to continue in different directions. This does not exactly reflect the real life situation, since sometimes tracks can split at a junction in between two stations as well. As a result, two infrastructure pieces might overlap.

However, since we do not have data on the exact locations of these splits and the lengths of the infrastructure pieces in between them in the appropriate format and the exact locations of these splits are not relevant for testing performance of the models, this is not taken into account in the current study. When applying the described approach, the covered infrastructure ends up being subdivided in 104 individual pieces, with an average length of 24.76 kilometers per piece. In Figure 1, we give an example of how the network is split. This figure shows part of the railway network around Utrecht and Amersfoort, where every dot represents a station. All stations that are colored black represent stations at which tracks split into different directions. Therefore, infrastructure pieces are split at these stations.

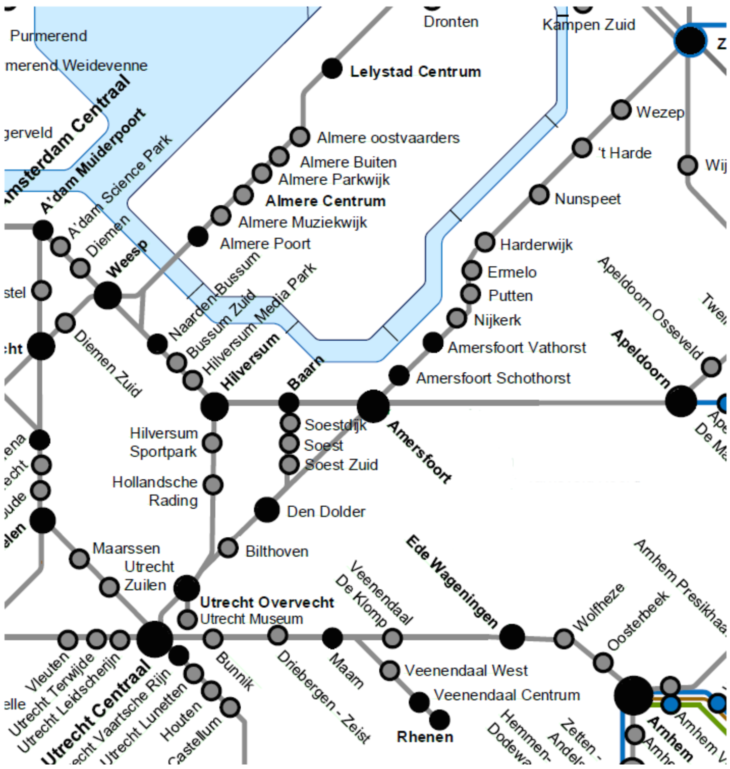

Figure 1: Division into infrastructure pieces

6.2.1 Variation Norm Values

When applying the basic model to the single-day instance, 150 out of the 2000 available duties are selected. In this solution, the number of unique kilometers for a driver based at Utrecht is equal to 251 and for a driver based in Amersfoort this value is equal to 245. The average variation is equal to 247.96. These values are much lower than the norm values mentioned in Section 2.3. However, since we only consider all tasks covered by Utrecht in one day instead of one week and assign them to two depots, this was to be expected.

For this set of duties, different norm values are applied. First, we consider several instances in which a certain norm value is set for both crew depots. Next, several instances are considered in which we differentiate between the norm values of the two depots. Finally, to test performance of the model including both variation norms and the average variation norm, we determine the average norm value. The norm values that will be considered are given in Table 6.3.

For the multi-day instance, the basic model yields a solution with 259.94 unique kilometers per driver for Utrecht and 245.03 for Amersfoort. The average variation is equal to 252.49. Therefore, we test the performance of the methods using norm value 300 for both Utrecht and Amersfoort and 300 for the average variation. Additionally, we apply the methods to the case in which the norm value for both depots is equal to 400, while the average norm value is equal to 450.

Table 6.3: Norm values real world instances

Norm value Utrecht	Norm value Amersfoort	Average norm value
250	250	-
260	260	-
270	270	-
260	250	-
275	225	-
300	200	-
250	250	250
260	260	275
270	270	300
250	200	250
260	250	255
275	225	250
300	200	250

7 Results

In this section, we present the results of the different methods obtained with the respective datasets. First, in Section 7.1 we determine the scoring method applied in the greedy heuristic for both formulations for the problem in which variation norms are included. Next, we discuss results for both methods on the artificial datasets and compare these to the solutions of the basic model (Section 7.2), followed by the results for the instances based on real life data (Section 7.3). For both instances we split the results in two parts. First, we only consider the variation restrictions per depot. Next, we add the average variation restriction to the problem.

7.1 Scoring Methods

The method applied to compute the scores of the different duties in the greedy algorithm has a large influence on the final solution produced by the algorithm. To compare the different scoring methods (described in Section 5.2), we have applied these to the artificial instance containing 200 tasks. This dataset was selected based on two criteria. Firstly, the behavior of the algorithms on smaller instances might not be representative of that on larger instances. Secondly, as running time increases with the size of the instance, testing the different score calculations on larger datasets is time consuming.

For each method we evaluate, besides the original score calculation, three different score calculations, each of which is applied with different parameter values. To evaluate the respective methods, we report the objective function of the final solution, the number of duties selected and the number of kilometers deviation from the norm for all depots combined and the corresponding penalty costs. The penalty cost parameter is set to 50 per unit deviation from the norm per depot. The norm value is set to 250 for all depots in order to test the effect of the scoring methods when the restrictions are not easily satisfied.

7.1.1 Method 1: Linearize Floor Function

For the method in which the floor function is linearized exactly, we first apply the original scoring method $\sigma_{sj} = \frac{\gamma_{sj}}{\mu_{sj}}$. This method only considers how many uncovered tasks are covered by a duty and its costs, but does not take into account its effect on variation of the corresponding depot. The other three scoring methods are applied with different values of parameter β.

Table 7.1: Different score computations Method 1

ID	Formula	β	Objective	Duty costs	Penalty costs	Duties	Time (s)
(i)	$\frac{c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t}{\mu_{sj}}$		38793.00	31168.00	7625.00	58	39.20
(ii)	$\frac{c_{sj}}{\mu_{sj}} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s$	1	38306.75	35213.00	3093.75	66	50.35
		10	38306.75	35213.00	3093.75	66	45.93
		50	38306.75	35213.00	3093.75	66	47.48
(iii)	$\frac{c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s}{\mu_{sj}}$	1	38078.38	32719.00	5359.38	61	34.04
		10	38078.38	32719.00	5359.38	61	38.11
		50	38078.38	32719.00	5359.38	61	38.89
(iv)	$\frac{c_{sj}}{\mu_{sj} + \sum_{t \in T_j \cap M^*} \lambda_t + \beta\theta_s}$	1	41063.13	37735.00	3328.13	71	55.31
		10	41063.13	37735.00	3328.13	71	54.29
		50	41063.13	37735.00	3328.13	71	57.03

From Table 7.1, we observe that penalty costs are relatively high when the original score calculation (i) is applied, making up nearly 20 percent of total costs. Applying calculation (ii) reduces penalty costs, but increases the number of duties selected. However, total costs decrease compared to the original scoring method. When scores are calculated using (iii), the decrease in penalty costs is lower than for (ii), but the increase in selected duties is also less steep. As a result, total costs are slightly lower than when applying (ii). Finally, calculation (iv) does reduce penalty costs by over 50 percent, but leads to an increase in number of selected duties of around 20 percent. Since duty costs increase considerably, total costs are higher than in the case of the original scoring method.

It can be observed that for all considered alternatives, the value of parameter β does not affect the resulting solution. However, there are some slight differences in running time.

Overall, it can be concluded that score calculation (iii) yields the best results, since penalty costs are reduced without drastically increasing the number of duties selected. Overall, this method gives a good balance between both aspects of our objective function. Therefore, this method, with parameter value $\beta = 1$, is applied in obtaining further results for Method 1.

7.1.2 Method 2: Approximate Floor Function

For the method in which we use a lower bound on the value of the floor function, we again first apply the original scoring method. Next, the different scoring methods are applied, with different values of parameter β. Table 7.2 shows some statistics of the resulting solutions, namely total costs, which is split up into duty costs and penalty costs, the number of duties selected and running time of the algorithm.

Table 7.2: Different score computations Method 2

ID	Formula	β	Objective	Duty costs	Penalty costs	Duties	Time (s)
(i)	$\frac{c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t}{\mu_{sj}}$		45791.38	29307.00	16484.38	55	349.55
(ii)	$\frac{c_{sj}}{\mu_{sj}} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s$	1	43588.13	27385.00	16203.13	51	365.01
		10	43588.13	27385.00	16203.13	51	459.53
		50	43568.00	27443.00	16125.00	51	551.10
(iii)	$\frac{c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \beta\theta_s}{\mu_{sj}}$	1	51810.50	28998.00	22812.50	54	468.13
		10	55285.38	29176.00	26109.38	54	380.38
		50	54846.88	50300.00	4546.88	94	1466.72
(iv)	$\frac{c_{sj}}{\mu_{sj} + \sum_{t \in T_j \cap M^*} \lambda_t + \beta\theta_s}$	1	61590.50	28403.00	33187.50	53	435.22
		10	61590.50	28403.00	33187.50	53	443.98
		50	44893.88	35597.00	9296.88	67	500.19

When we apply original score calculation (i), 55 duties are selected and penalty costs are around 35 percent of total costs. Applying score calculation (ii) reduces costs, mainly resulting from a decrease in duty costs, but penalty costs also reduce slightly. Hence, taking variation constraints into account when selecting duties can even lead to more efficient schedules. The value of parameter β only has a small effect on the objective value, but has a larger effect on running time. Since running time is considerably higher for higher values of β, the most appropriate value of β seems to be 1. For score calculation using (iii), results differ tremendously for different parameter values. For $\beta = 1$ and $\beta = 10$ duty costs remain more or less equal, but penalty costs actually increase largely. On the contrary, $\beta = 50$ yields a solution with very low penalty costs, but selects 70 percent more duties than the original scoring method. A similar pattern can be observed for the final score calculation (*iv*), since $\beta = 1$ and $\beta = 10$ both yield solutions with high penalty costs, but $\beta = 50$ actually leads to relatively few violations.

Since calculation (*ii*) produces the best results with respect to overall objective value and gives stable results for different values of β, this method is applied for the remaining results for Method 2. Since running time increases considerably for larger values of β without positively affecting the solution quality, this parameter is set equal to 1.

When we recalculate the actual costs using the true variation measure, total costs for this formulation method can reduce, since the actual constraint is less strict than the one on which the penalty costs are based. Therefore, penalty costs reported in Table 7.2 can be higher than the actual penalty in case this solution would be applied. However, duty costs remain equal, so in case high costs are largely due to selecting many duties, this effect is small.

7.2 Artificial Instances

In this section we will present the results obtained by applying the different methods on the artificial instances. First, we will consider the basic problem in which no variation restrictions are present. These results can be used as a reference to evaluate the effect of adding variation norms on the objective values and selected duties. Next, we will consider the case in which only variation norms for the different depots are present. Finally, we will add the average variation norm over all depots and analyze the corresponding results.

7.2.1 Basic Problem

First, we consider the problem in which all tasks need to be assigned to duties, but no variation restrictions are present.

Optimal Solution When only set covering restrictions are present, we are able to find the optimal solution to some of the instances. These results are reported in Table 7.3. For the instance with 400 tasks, the optimality gap is just below 10 percent after 2 hours, indicating how rapidly running time increases with problem size.

Table 7.3: Optimal solutions for artificial instances without variation norms

Tasks	Objective	Duties	Time (s)
50	13207.00	25	0.08
100	11741.00	22	0.19
200	23470.00	43	2.20
300	34263.00	63	53.88

Heuristic Even for small instances the set covering problem without variation norms is difficult to solve. Therefore, we apply the basic algorithm, as described in Section 5.1, and compare the results. Furthermore, having these results allows analyzing the effect of variation norms on the performance of the heuristic method.

We display the results of the heuristic method for the basic algorithm on the artificial instances in Table 7.4. It can be observed that for the small instances a few more duties are selected than in the optimal solution, for example 24 instead of 22 for the instance with 100 tasks. This leads to an increase in objective value of roughly between 4 and 12 percent. In addition, it can be observed that for the instances for which we are able to find the optimal solution, the running time of the heuristic method exceeds that of the optimal solution. However, running time of the heuristic method increases less rapidly with size of the instance than that of finding the optimal solution, hence we are able to find feasible solutions for larger instances as well.

Table 7.4: Heuristic solutions for artificial instances without variation norms

Number of tasks	Objective	Duties	Time (s)	Optimality gap (%)
50	13848.00	26	1.64	4.85
100	13041.00	24	3.05	11.07
200	24719.00	46	20.37	5.32
300	38041.00	70	128.35	10.18
400	46072.00	85	325.47	-

7.2.2 Variation Norm per Depot

In this section we describe the effect of including variation norms in the model. We apply the norm values that are reported in Table 6.2.

First, we will consider the optimal solution to some instances of this problem. Afterwards, we apply the basic model and both methods for modeling the variation measure and compare the results.

Optimal Solution Since this problem is difficult to solve to optimality, we report the optimal solution only for some small instances. These results are given in Table 7.5. It can be observed that the objective values only increase slightly. For both the instances with 100 and 200 tasks, the the number of selected duties has increased by one. Furthermore, it can be observed that running time increases significantly compared to the case in which no variation restrictions are present. For the instance with 300 tasks, we are no longer able to find the optimal solution within reasonable time.

Table 7.5: Optimal solutions for artificial instances with variation norm per depot

Tasks	Norm	Objective	Duties	Time (s)
50	42	13207.00	25	0.18
100	40	12452.00	23	33.20
200	100	23846.00	44	400.01
300	30	-	-	-

Method 0: Basic Model One way to obtain a solution for the problem with variation norm per depot is to use Method 0, in which penalty costs are added to the solutions reported in Table 7.4 for the constraints that are not satisfied. The resulting solution values are reported in Table 7.6, where 'duty costs' gives the objective value from Table 7.4 and penalty costs are calculated based on the variation present in the corresponding solution. It can be observed that the optimality gap has increased compared to the case in which we do not include variation

restrictions. This is due to the fact that duty costs remain equal, but penalty costs are now added for violations of the restrictions, while the effect of the variation norms on the objective value of the optimal solution is limited.

Table 7.6: Results Method 0 for artificial instances with variation norm per depot

Tasks	Norm	Objective	Duty cost	Penalty cost	Duties	Time (s)	Optimality gap (%)
50	42	14148.00	13848.00	300.00	26	1.64	7.13
100	40	17541.00	13041.00	4500.00	24	3.05	40.15
200	100	29719.00	24719.00	5000.00	46	20.37	24.63
300	30	54541.00	38041.00	16500.00	70	128.35	-
400	30	56571.00	46072.00	10500.00	85	325.47	-

Method 1: Linearize Floor Function Next, we test Method 1 in which variation is incorporated in the solution approach. Variation norm values given in Table 6.2 are applied. The resulting objective values, split up into duty costs and penalty costs, the number of duties selected and running time of the algorithm are given in Table 7.7.

Table 7.7: Results Method 1 for artificial instances with variation norm per depot

Tasks	Norm	Objective	Duty cost	Penalty cost	Duties	Time (s)	Optimality gap (%)
50	42	14611.00	14611.00	0.00	27	5.92	10.63
100	40	14427.00	14427.00	0.00	27	19.00	15.86
200	100	26429.00	26429.00	0.00	49	77.90	10.83
300	30	49006.00	40006.00	9000.00	74	175.71	-
400	30	55237.00	46237.00	9000.00	86	248.47	-

For the datasets for which the optimal solution is known (see Table 7.5, the number of selected duties in the heuristic solution is slightly higher than in the optimal solution. As a consequence, duty costs and hence total costs increase, while penalty costs remain 0. Running time increases less rapidly with size of the dataset than for the optimal solution.

When we compare the results of this method to those of Method 0, reported in Table 7.6, we observe that, except for the dataset with 50 tasks, objective values decrease. A few more duties have been selected for all datasets, hence duty costs increase, but penalty costs are either reduced or eliminated. As a consequence, the optimality gap for the instances with 100 and 200 tasks is much smaller than when applying Method 0. Running times of the current solution method increase slightly compared to those of Method 0 for most instances.

Method 2: Approximate Floor Function We continue by applying Method 2 to the same instances, again using the norm values given in Table 6.2. The results are given in Table 7.8. When this method is applied, penalty costs are present when the variation restriction using $x-1$ instead of $\lfloor x \rfloor$ is not satisfied. However, since these constraints are more strict than the original ones, penalty costs are overestimated. Therefore, penalty costs and total costs are reevaluated using the original constraints. These values are given in the two columns 'Actual penalty' and 'Actual objective' of Table 7.8.

Table 7.8: Results Method 2 for artificial instances with variation norm per depot

Tasks	Norm	Objective	Duty cost	Penalty cost	Duties	Time (s)	Actual penalty	Actual objective	Optimality gap (%)
50	42	17535.00	17535.00	0.00	33	183.53	0.00	17535.00	32.77
100	40	16920.00	16920.00	0.00	32	776.68	0.00	16920.00	35.88
200	100	28887.00	28887.00	0.00	54	6540.79	0.00	28887.00	21.14
300	30	51529.75	46411.00	5118.75	87	19231.47	5118.75	51529.75	-
400	30	52951.50	47364.00	5587.50	89	8551.98	900.00	48264.00	-

Similar to Method 1, the actual objective value, evaluated using the correct variation norm, is lower than that of Method 0 for all instances except for the one with 50 tasks. However, the objective values are higher than those of Method 1. This can also be observed from the larger optimality gaps. Since for the datasets with 50, 100 and 200 tasks a solution without penalty costs has been found, the actual objective value is equal to that calculated by the algorithm. The selection of duties for these datasets satisfies the stricter variation norms, hence the objective value is higher.

We observe that running times for the current method are higher than those of Method 0 and Method 1. This may be due to the fact that more duties need to be added to satisfy the variation norms and hence more time is needed to obtain a feasible solution to the problem.

Comparison Based on the previous results, it can be concluded that incorporating variation norms per depot in the solution approach can lead to lower cost solutions without drastically increasing computation times, when Method 1 is applied. This can be observed from the fact that the optimality gap for Method 1 is smaller than for Method 0 on most instances when we include variation restrictions. In addition, the effect on the number of selected duties is limited. Method 2 also produces solutions with lower objective values than Method 0, though often with slightly higher values than those of Method 1, but requires large computation times as the size of the dataset increases.

7.2.3 Variation Norm per Depot and Average Variation Norm

We continue by analyzing the results of the extended problem, in which a restriction on the average variation over all depots is added.

Optimal Solution For some of the small test instances, we are able to find an optimal solution. These are reported in Table 7.9. It can be observed that adding the average variation norm to the problem does not change the optimal solution for the instances with 50 and 100 tasks compared to the case in which we only consider the variation restriction per depot. For the instance with 200 tasks the objective value increases slightly. Higher levels of the norm value for the average variation restriction are expected to lead to solutions in which more duties are selected and hence costs are higher. We do observe an increase in running time for the instance with 200 tasks.

Method 0: Basic Model In order to compare performance of the methods that incorporate variation norms, we again first calculate the solution value of Method 0. We calculate penalty costs based on the variation at the individual depots an the average variation and add these

Table 7.9: Optimal solutions for artificial instances with variation norm per depot and average variation norm

Tasks	Norm	Avg norm	Objective	Duties	Time (s)
50	42	45	13207.00	25	0.14
100	40	70	12452.00	23	27.23
200	100	250	23853.00	44	763.24
300	30	100	-	-	-

costs to the solution value of the algorithm that does not consider variation. Corresponding duty costs, penalty costs and objective values are given in Table 7.10. Again, it can be observed that the optimality gaps for the different instances have increased compared to Section 7.2.1, where no variation restrictions were considered. Especially for the instances in which penalty costs are large, the optimality gap is large.

Table 7.10: Results Method 0 for artificial instances with variation norm per depot and average variation norm

Tasks	Norm	Avg norm	Objective	Duty cost	Penalty cost	Duties	Time (s)	Optimality gap (%)
50	42	45	14363.50	13848.00	515.50	26	1.64	8.76
100	40	70	17993.00	13041.00	4952.00	24	3.05	44.50
200	100	250	31169.00	24719.00	6450.00	46	20.37	30.67
300	30	200	55149.50	38041.00	17108.50	70	128.35	-
400	30	280	56988.00	46072.00	10916.00	85	325.47	-

Method 1: Linearize Floor Function We will now discuss the results when we apply Method 1 to the artificial instances, in which we include both variation norms per depot and the average variation norm. The average norm value is slightly higher than the average variation in the basic model, hence the instance is somewhat restricted. The resulting objective values, duty costs, penalty costs and number of selected duties are given in Table 7.11 for the different artificial instances. In addition, running times of the algorithm are reported.

Table 7.11: Results Method 1 for artificial instances with variation norm per depot and average variation norm

Tasks	Norm	Avg norm	Objective	Duty cost	Penalty cost	Duties	Time (s)	Optimality gap (%)
50	42	45	15140.57	15137.00	3.57	28	6.29	14.64
100	40	70	14938.00	14938.00	0.00	28	19.20	19.96
200	100	250	26390.70	25679.00	711.70	47	67.99	10.63
300	30	200	49737.00	40737.00	9000.00	75	177.55	-
400	30	280	55203.00	46203.00	9000.00	85	244.77	-

As a consequence of the additional restriction, objective values have increased slightly for most instances, due to increased duty costs and/or increased penalty costs. It can be observed that for the datasets with 200 and 400 tasks the objective value has actually decreased compared to the case where we apply Method 1 with variation norms per depot only. This is counterintuitive, since the problem is more restricted, but is caused by the fact that a heuristic does not yield optimal solutions. It may be the case that the optimality gap of a more restricted problem is smaller than for the less restricted problem, which can cause the objective value of the more restricted problem to be lower.

Similar to the case in which Method 1 is applied to the problem with variation norms per depot only, we obtain lower objective values than with Method 0 for all instances except the one with 50 tasks, while the increase in the number of duties selected is limited. This is also reflected in the optimality gaps of the small instances. Except for the instance with 50 tasks, the optimality gap is considerable smaller than using Method 0.

Method 2: Approximate Floor Function Next, we apply Method 2 to the problem in which both types of variation norms are included. Again, we can observe that the solution for the dataset with 50 tasks has a worse objective value than Method 0. Since no penalty costs are included in this solution, actual costs (using the actual variation norm instead of approximated) are equal to the objective value. For the dataset with 100 tasks, we do obtain an improved solution compared to Method 0. This solution does not contain penalty costs, but the number of selected duties has increased by one compared to the problem in which variation norms per depot were included only. When we consider the solutions for the datasets with 200, 300 and 400 tasks, we can observe the difference between the objective value of the resulting solution and the actual objective value. The objective values given by the method slightly exceed the solution values of Method 0, whereas the actual objective values are below the solution values of Method 0. This is also reflected by the optimality gaps for the instances with 100 and 200 tasks, as these gaps are slightly below those of the solution produced by Method 0.

Interestingly, running times for this method when including average variation restrictions are lower than when we do not include these restrictions. However, running times for Method 2 do exceed those of Method 1.

Table 7.12: Results Method 2 for artificial instances with variation norm per depot and average variation norm

Tasks	Norm	Avg norm	Objective	Duty cost	Penalty cost	Duties	Time (s)	Actual penalty	Actual objective	Optimality gap (%)
50	42	45	18542.00	18542.00	0.00	35	37.02	0.00	18542.00	40.39
100	40	70	17569.00	17569.00	0.00	33	92.37	0.00	17569.00	41.09
200	100	250	31915.84	30429.00	1486.84	57	1871.46	365.79	30794.79	29.10
300	30	200	58293.67	38094.00	20199.67	71	7864.52	13633.19	51757.19	-
400	30	280	60141.14	46863.00	13278.14	88	7424.02	10765.57	57628.57	-

Comparison Similar to the case where we included variation norms per depot only, including the variation constraints in the solution approach yields better solutions than Method 0 in terms of total costs. Higher variation levels can be achieved by selecting several additional duties. As a consequence the optimality gap for Method 1 and Method 2 is smaller than that for Method 0 on all instances except for the instance with 50 tasks. Again, Method 1 seems to perform better than Method 2 both in terms of costs and running time. Running times of Method 1 for the case in which we include variation norms per depot as well as an average variation norm are similar to those in case only variation norms per depot are considered.

7.3 Real Life Instances

In this section, we present results of the developed methods applied to test instances created based on real life data. We start with applying the different methods to the single-day real life instance. We apply Method 0, such that we can analyze the effect of including variation norms. First, we only include variation norms per depot to the model (Section 7.3.2), after which we also include an average variation norm over all depots (Section 7.3.3). Finally, we will present some results regarding the multi-day real life instance in Section 7.3.4.

7.3.1 Basic Problem

When applying the basic algorithm (without considering variation norms) to the single-day real life instance, 150 duties are selected out of the 2000 available duties, resulting in costs equal to 80840.00. The running time of the algorithm is 1295.25 seconds. In this solution, the unique number of kilometers for a driver based at Utrecht is equal to 251 and for a driver based in Amersfoort this value is equal to 245. Average variation is equal to 247.96.

7.3.2 Variation Norm per Depot

In this section, we present the results for the different methods for solving the problem in which we incorporate variation norms per depot only. First, we will consider the solutions of the problem that does not take into account these restrictions, after which we will discuss both developed methods.

Method 0: Basic Model First, we present the solution of Method 0, including penalty costs for violations of the variation restrictions. These results are reported in Table 7.13 and can be used as a benchmark for the other methods. Duty costs concern the costs of the selected duties by the basic model, which are equal for the different norm values. Penalty costs are calculated for the different norm values based on the variation present in the solution obtained using the basic model.

Table 7.13: Results Method 0 real instance with different variation norms per depot

Norm Ut	Norm Amf	Objective	Duty costs	Penalty costs	Duties	Time (s)
250	250	81090.00	80840.00	250.00	150	1295.25
260	260	82040.00	80840.00	1200.00	150	1295.25
270	270	83040.00	80840.00	2200.00	150	1295.25
260	250	81450.00	80840.00	700.00	150	1295.25
275	225	82040.00	80840.00	1200.00	150	1295.25
300	200	83290.00	80840.00	2450.00	150	1295.25

Method 1: Linearize Floor Function We will now consider the results of applying Method 1 to the single-day real life instance, with the same variation norms. To check whether the score calculation method based on the artificial instances is appropriate for the realistic instance, we test all three methods on the instance with norm value 260 for both depots. We use parameter

$\beta = 1$ for all three methods. The results are presented in Table 7.14. It can be concluded that *(ii)* and *(iii)* have similar performance, although *(iii)* produces a solution with lower penalty costs and total costs. The final method, *(iv)*, yields a solution with a considerably higher objective value than the other two methods, due to a large number of selected duties. Penalty costs for this solution are low, but the method required much more computation time than the other two. It can thus be concluded that the score calculation method selected based on the random instances is also appropriate for the instances based on real life data.

Table 7.14: Score computations Method 1 on real instance

ID	Formula	Objective	Duty costs	Penalty costs	Duties	Time (s)
(ii)	$\frac{c_{sj}}{\mu_{sj}} - \sum_{t \in T_j \cap M^*} \lambda_t - \theta_s$	83268.00	80768.00	2500.00	149	1142.53
(iii)	$\frac{c_{sj} - \sum_{t \in T_j \cap M^*} \lambda_t - \theta_s}{\mu_{sj}}$	82635.00	81635.00	1000.00	150	825.77
(iv)	$\frac{c_{sj}}{\mu_{sj} + \sum_{t \in T_j \cap M^*} \lambda_t + \theta_s}$	85143.00	85017.00	126.19	159	2623.61

We continue by applying Method 1 using score calculation method *(iii)* to the instance based on real life data with the different norm values. These results are presented in Table 7.15. It can be observed that the objective value resulting from Method 1 is slightly above that of Method 0 for most instances. Since more or different duties are selected, duty costs are higher than in case variation was not considered. However, in general penalty costs have decreased. When we consider norm value 300 for Utrecht and 200 for Amersfoort, we observe that the number of duties actually decreases compared to Method 0. Still, more variation is realized by selecting these duties, as penalty costs decrease. From the last column, we can observe that running times for Method 1 are lower than for Method 0.

Table 7.15: Results Method 1 real instance with different variation norms per depot

Norm Ut	Norm Amf	Objective	Duty costs	Penalty costs	Duties	Time (s)
250	250	81485.00	81485.00	0.00	150	763.43
260	260	82635.00	81635.00	1000.00	150	825.77
270	270	84820.29	82648.00	2172.29	153	935.54
260	250	82539.00	82539.00	0.00	153	717.49
275	225	83234.59	82014.00	1220.59	151	795.13
300	200	82891.00	81041.00	1850.00	147	763.14

Method 2: Approximate Floor Function When we apply Method 2 to the single-day real life instance with variation norm per depot, we first of all observe that large running time is required. When we use norm values 250 for both depots, running time is equal to 11433.61 seconds. The objective value is equal to 83826.55, consisting of 80772.00 duty costs and 3054.55 penalty costs. When we reevaluate this solution using the actual variation measure, penalty costs are slightly lower, namely 3031.25, leading to total costs 83803.25. Even though duty costs are lower than those of both other methods, total costs are higher due to higher penalty costs. Since running times for this method are impractically large and it does not clearly outperform the other methods, we do not investigate this method further on the real life instances.

Comparison First of all, Method 2 turned out to require too much computation time to be of practical use on the realistic instance. Next, we compare Method 1 to Method 0. It is observed that the objective value of Method 1 is slightly higher than for Method 0, due to higher duty costs. However, penalty costs are lower, while the increase in number of selected duties is limited. In addition, computation time for Method 1 is actually lower than for Method 0. Based on these observations, it can be concluded that Method 1 can be applied to obtain solutions with more variation without increasing computation time. Furthermore, the produced solution includes only slightly more duties than the solution of Method 0, resulting in somewhat higher duty costs. For the final instance, with norm values 300 and 200 for Utrecht and Amersfoort and 250 average, the number of selected duties has actually decreased compared to the solution of Method 0, even though duty costs have increased.

7.3.3 Variation Norm per Depot and Average Variation Norm

We continue by applying both Method 0 and Method 1 to the problem in which we incorporate both variation restrictions per depot and the average variation restriction over all train drivers.

Method 0: Basic Model To compare performance of Method 1 to the basic solution, we compute penalty costs for this solution using different norm values. In Table 7.16, we present objective values, duty costs and penalty costs for different values of the variation norms. In the first rows, the same norm value is applied to both depots, whereas in the last rows we differentiate between the two depots.

Table 7.16: Results Method 0 real instance with different variation norms per depot and average variation norms

Norm Ut	Norm Amf	Avg norm	Objective	Duty costs	Penalty costs	Duties	Time (s)
250	250	250	81192.00	80840.00	352.00	150	1295.25
260	260	275	83392.00	80840.00	2552.00	150	1295.25
270	270	300	85642.00	80840.00	4802.00	150	1295.25
250	200	250	80942.00	80840.00	102.00	150	1295.25
260	250	255	81892.00	80840.00	1052.00	150	1295.25
275	225	250	82142.00	80840.00	1302.00	150	1295.25
300	200	250	83392.00	80840.00	2552.00	150	1295.25

Method 1: Linearize Floor Function We apply Method 1 to the single-day instance with the same norm values as given in Table 7.16. First, we consider the instances in which the same norm value is applied for both Utrecht and Amersfoort. These results are given in the first rows of Table 7.17. It can be observed that the objective values are similar to those of Method 0, but exceed those slightly. However, penalty costs did decrease compared to Method 0, at the cost of some more duties.

Similar observations can be made for the bottom part of the table, in which we differentiate between the norm values for the two depots. In some of these instances Method 1 slightly outperforms Method 0, while in others it does not. We observe that costs increase as we increase the norm value for one of the depots (Utrecht), due to increasing penalty costs and duty costs. Decreasing the norm value for the other depot, for example 275 for Utrecht and 225 for Amersfoort, does not lead to lower variation for the other depot. This may be due to the fact that the average variation constraint needs to be satisfied and that in the basic solution variation is quite similar at both depots.

In all cases, applying Method 1 leads to lower penalty costs than applying Method 0. However, since penalty costs are used to quantify the consequences of not satisfying the variation norms, higher total costs represent worse solutions. Nonetheless, the difference in objective values is limited.

Table 7.17: Results Method 1 real instance with different variation norms per depot and average variation norms

Norm Ut	Norm Amf	Avg norm	Objective	Duty costs	Penalty costs	Duties	Time (s)
250	250	250	82045.00	82045.00	0.00	152	678.88
260	260	275	84529.99	82198.00	2331.99	151	814.71
270	270	300	87663.05	83562.00	4101.05	156	1079.34
250	200	250	81094.95	81088.00	6.95	151	1011.68
260	250	255	81257.63	80653.00	604.64	151	775.08
275	225	250	82369.00	81669.00	700.00	152	1166.22
300	200	250	85297.00	83497.00	1800.00	156	1215.73

Comparison Similar to the case in which we only included variation norms per depot on the single-day real instance, Method 1 has slightly higher objective values than Method 0. In addition, we again observe a slight increase in the number of duties selected and a decrease in penalty costs, when we apply Method 1 instead of Method 0.

Sensitivity

Penalty Cost Parameter We continue by analyzing the effect of a higher penalty cost parameter on the difference between the objective values. We apply both methods again to the same instance with the same norm values, but use penalty costs 100 for each kilometer below the norm value. The resulting solutions for Method 0 and Method 1 are given in Table 7.18 and Table 7.19, respectively.

Similar to the case in which we use penalty value 50, objective values of both methods are not that different. The decrease in penalty costs when applying Method 1 instead of Method 0 is even more pronounced, but so is the increase in duty costs for some of the instances. For other instances, for example with norm values 250 and 200 for Utrecht and Amersfoort and 250 average, penalty costs decreased compared to Method 0, while duty costs only increased slightly. Therefore, total costs did decrease compared to Method 0 and more variation was included in the schedules.

When we compare the output of Method 1 with penalty costs 50 and 100, we again observe that objective values are similar for both cases. In some cases, penalty costs 100 lead to considerably higher costs, for example with norm value 270 for both depots and average norm 300. This is due to the fact that violation of the constraints was not reduced, hence penalty costs are doubled. However, in general, changing the penalty value does not largely affect the objective value, but mainly changes the distribution between duty costs and penalty costs.

Table 7.18: Results Method 0 real instance with different variation norms per depots and average variation norms, penalty = 100

Norm Ut	Norm Amf	Avg norm	Objective	Duty costs	Penalty costs	Duties	Time (s)
250	250	250	81544.00	80840.00	704.00	150	1295.25
260	260	275	85944.00	80840.00	5104.00	150	1295.25
270	270	300	90444.00	80840.00	9604.00	150	1295.25
250	200	250	81044.00	80840.00	204.00	150	1295.25
260	250	255	82944.00	80840.00	2104.00	150	1295.25
275	225	250	83444.00	80840.00	2604.00	150	1295.25
300	200	250	85944.00	80840.00	5104.00	150	1295.25

Table 7.19: Results Method 1 real instance with different variation norms per depot and average variation norms, penalty = 100

Norm Ut	Norm Amf	Avg norm	Objective	Duty costs	Penalty costs	Duties	Time (s)
250	250	250	81004.00	81004.00	0.00	148	1317.45
260	260	275	86940.18	84832.00	2108.18	160	1343.72
270	270	300	91764.09	83562.00	8202.09	156	1462.52
250	200	250	80855.00	80855.00	0.00	149	1015.94
260	250	255	81176.00	81176.00	0.00	153	1023.78
275	225	250	83573.00	83573.00	0.00	153	1210.81
300	200	250	87097.00	83497.00	3600.00	156	1107.34

Update Number of Drivers As described in Section 5.2.4, we start with a fixed number of drivers and update it at the end of every iteration of the greedy heuristic. However, it is not clear what the initial value of the number of drivers for a certain depot should be. We tested different values for this parameter and analyzed the resulting solutions. It turned out that, for all tested instances, the initial value does not affect the final solution. Furthermore, there is an important update in the number of drivers after the first iteration of the greedy procedure, but for the following iterations the number of drivers per depot remains stable. This indicates that the initial value does not have a large effect on the final result.

7.3.4 Multi-Day Instance

In this section, we will present the results of both Method 0 and Method 1 on the real life instance with tasks for multiple days.

Method 0: Basic Model When we apply Method 0, the resulting solution has duty costs equal to 221885.00. Running time of the algorithm is equal to 2680.49 seconds. Out of the 6000 available duties, 407 duties are selected. This results in 259.94 unique kilometers for a train driver based at Utrecht and 245.03 for one based at Amersfoort. On average, the number of unique kilometers for a train driver is equal to 252.49.

Now, we apply different sets of norm values and calculate the penalty costs. These values, in combination with the resulting objective values are given in the first part of Table 7.20.

Table 7.20: Comparison Method 0 and Method 1 on multi-day instance

	Norm	Avg norm	Objective	Duty costs	Penalty costs	Duties	Time(s)
Method 0	300	300	228989.50	221885.00	7104.50	407	2680.49
	400	450	246489.00	221885.00	24604.50	407	2680.49
Method 1	300	300	198625.00	198625.00	0.00	362	1594.93
	400	450	199694.00	199694.00	0.00	364	1610.16

Method 1: Linearize Floor Function Next, we apply Method 1 to the instances with variation restrictions. These results are reported in the final part of Table 7.20.

For the first instance with variation norms per depot and average variation norm equal to 300, we obtain a solution with duty costs equal to 198625.00. In this solution no penalty costs are present, hence total costs are also equal to 198625.00. When we compare this value to the solution of Method 0 we actually observe a decrease in the number of duties selected and corresponding duty costs. Since also no penalty costs are present, Method 1 clearly outperforms Method 0.

When we consider the instance with norm value 400 per depot and 450 average, we again obtain a solution without penalty costs. The objective value of this solution is 199694.00 and 364 duties are selected. Hence, Method 1 again outperforms Method 0.

Comparison It can be concluded that Method 1 is also applicable to instances in which multiple days are considered. Running time does increase compared to the single-day instance, but is lower than that of Method 0. In addition, while the solution produced by Method 0 includes only slightly more variation than the single day instance, Method 1 can produce solutions with much more variation.

Furthermore, we can conclude that the performance of Method 1 in comparison to Method 0 on the instances with variation restrictions has improved. On the single-day instance, total costs were relatively equal and only limited increases in variation could be generated by Method 1. On the multi-day instance, more duties covering the different infrastructure pieces are available. Therefore, there are more possibilities to increase variation. It can therefore be concluded that the effect of incorporating the variation restrictions in the solution approach is larger when we consider a longer planning horizon.

8 Conclusions and Discussion

In this final section, we will present our conclusions and discuss to what extend we have been able to solve the research questions. In addition, we will present our advice to NS on the topic of incorporating variation restrictions in crew scheduling algorithms. Finally, suggestions will be made for further research.

8.1 Summary of Applied Methods

Every week many tasks need to be assigned to duties that are then assigned to the train drivers and guards at NS. Given the large number of possibilities and numerous labor rules that need to be considered, crew scheduling is a complex problem. Besides these complexities, NS wants to incorporate restrictions concerning variation in the schedule of train drivers and guards into the crew scheduling algorithms. Agreements with the workers council were made on how variation should be measured and what norm values should be satisfied. Due to the nonlinear nature of the measure for variation, incorporating these constraints is not straightforward. The aims of this study were 1) to investigate how these variation restrictions could be modeled, such that they could be incorporated in the set covering formulation of the crew scheduling algorithm and 2) to explore whether it is possible to adjust the current crew scheduling algorithm as to incorporate the constraints regarding average variation per train driver or guard.

Two methods were considered for formulating the variation restrictions. First, we considered a method in which the variation measure was linearized exactly. This required addition of several (integer) auxiliary variables and many constraints to assign correct values to these variables. This method incorporates the variation restrictions exactly, but the set covering problem becomes even more difficult to solve. Therefore, we also investigated a second method for modeling the variation restrictions. With this method we did not linearize the floor function in the variation measure, but approximated it by its lower bound, i.e. instead of $\lfloor x \rfloor$ we used $x - 1$. Again, multiple additional variables and constraints need to be added to deal with remaining nonlinearities. The resulting problem was expected to be easier to solve, but to lead to higher objective values, since the constraints are more strict as variation is underestimated.

The next step was adjusting the solution approach for solving the resulting problem. NS currently applies an algorithm based on column generation and Lagrangian relaxation to solve the set covering problem. In this study it was considered how the set covering problem could be solved assuming a given subset of columns, produced in the column generation phase. It was considered how the current algorithm, based on the method introduced by Caprara et al. (1999), could be adjusted to incorporate the variation constraints. We developed two extended versions of the basic algorithm, based on the two suggested formulations for the crew scheduling problem.

Different test instances were created to test performance of the two methods. First, artificial test instances based on random data were used. Several datasets with varying numbers of tasks, duties and depots were created. Second, two more realistic instances created from actual data of NS were constructed. For the artificial instances, the levels for the variation norms were determined based on the variation present in the solution of the basic model. The norm values were set slightly above these values, such that the constraints restrict the problem, but do allow for useful solutions to still exist. For the real instances, the same approach was applied, but multiple norm values were considered.

To evaluate the performance of the methods and compare these to the algorithm in which no variation restrictions are considered, we calculated the actual objective value, in which we penalized violations of the restrictions, of the solutions of all three models (Method 0, Method 1 and Method 2). For the solution of Method 0, we calculated the level of variation present in the solution of the algorithm that did not incorporate variation and added penalty costs if the norm values were not satisfied. The solution of the approach that used the first formulation (Method 1), in which the variation measure was linearized exactly, already includes penalty costs for not satisfying the constraints. Finally, the solution of the algorithm based on the second formulation (Method 2) contains penalty costs for violating the constraints, but these are calculated using the approximated version of these constraints. To be able to compare the costs of this method to the other methods, we reevaluated the level of variation in this solution and calculated penalty costs based on these values.

8.2 Conclusions and Recommendations

Two new formulations for the crew scheduling problem with variation restrictions were proposed and compared to the algorithm in which variation is not considered. The variation restrictions were modeled differently in both formulations, one being an exact representation of the constraints and the other an approximation. Therefore, it can be concluded that it is possible to include the restrictions in the mathematical formulation of the crew scheduling problem.

Next, we evaluated the performance of the adjusted solution approach based on these two new formulations. We compared the resulting solutions to those of Method 0, which represents the solution in which variation was not incorporated in the algorithm, but penalty costs were added afterwards. For most artificial instances, both for the case in which we consider variation restrictions per depot only and in case we also consider the average variation restriction, Method 1 yields better solutions in terms of objective value than Method 0. As a consequence, we observe a smaller optimality gap for Method 1 than for Method 0 on the small instances. This is at the costs of a few more duties. Method 2 also gives lower cost solutions than Method 0 in most instances, but is outperformed by Method 1. In addition, applying Method 2 leads to the addition of more duties than applying Method 1. When we consider running times, Method 1 requires slightly more time than Method 0. As the size of the problem instance increases, the increase in running time for Method 1 is similar to that for Method 0. For Method 2, running time is larger than for both other methods. In addition, running time increases more rapidly as the size of the instance increases.

Performance of Method 1 on the single-day real life instance is similar to that on the artificial instances. On this instance, for the different norm values, Method 1 yields solutions similar to those of Method 0 in terms of objective value. For most norm values (both with and without average variation norm), Method 1 has slightly higher objective values than Method 0, but penalty costs are between 15 and 100 percent smaller. Since between 0.5 and 4 percent more duties are selected, duty costs are up to 5 percent higher. When we differentiate norm values for the two depots, the results remain similar. Setting a high norm value for one depot and a lower one for the other does not automatically lead to the second depot actually having less variation. This is probably due to the fact that the variation levels for the two depots in the solution of the basic model are very similar. When we applied Method 2 to this instance, we observed considerably larger running times, while the method did not outperform the other two methods. As a consequence, it was concluded that it is impractical to apply Method 2 to the real life instances. On the multi-day instance, Method 1 clearly outperformed Method 0, as it was able to find solutions with nearly 80 percent more variation that actually yielded lower duty costs than the solution of Method 0. This shows that when there are more possibilities to include variation in the schedule, Method 1 clearly utilizes these and outperforms Method 0 that does not consider variation.

Furthermore, we investigated the effect of adjusting the value of the penalty parameter for Method 1 on the single-day real life instance with both variation restrictions per depot and the average variation restriction. In general, setting higher penalty for violations leads to more duties being selected and a decrease of violation. As a result, penalty costs actually decrease, but duty costs increase. Therefore, the effect on the objective value is small. However, if it is not possible to reduce violation further, penalty costs and hence objective value increase, since each violation is penalized more heavily.

Overall, it can be concluded that both Method 1 and Method 2 can be applied to generate solutions while taking variation restrictions into account. Method 1 in general has objective values that are similar to those of Method 0. When we look at the number of duties, Method 1 only selects a limited number of additional duties as compared to Method 0. In addition, it was observed that the performance of Method 1 in comparison to Method 0 improves considerably when the planning horizon increases and more possibilities for variation are available. For Method 2, the objective value is often higher than that of Method 0. However, this value is calculated using the more strict variation constraints in which the floor function is approximated. When these costs are high due to large penalty costs, actual costs when evaluated using the correct variation measure may be lower. However, when many duties are selected to satisfy these strict variation norms, the difference between the objective value and the actual costs may be much smaller. Since the division between penalty costs and duty costs is unpredictable, we cannot give a clear indication of the actual costs of Method 2. Since the algorithm aims to satisfy more strict variation norms, more duties are selected in comparison to Method 1 and Method 0.

Based on these conclusions, our recommendation is to apply Method 1 for incorporating variation restrictions in the crew scheduling algorithm. In terms of costs, the performance of Method 1 is similar to that of Method 0. Furthermore, variation can be increased without drastically increasing the total number of duties. Especially, since crew scheduling at NS is considered on a weekly basis and hence many possibilities for variation are present, this method is expected to outperform Method 0.

8.3 Limitations

Certain choices made when designing the methods evaluated in this study may pose limitations that need to be taken into account when interpreting the results.

When we adjusted the greedy algorithm to incorporate variation restrictions, we considered several different scoring methods to determine which duties are added. For both Method 1 and Method 2 we considered three different formulas, each with three different parameter values. However, there might be different formulas or parameter values that lead to better solutions.

Furthermore, in the *Subgradient phase*, we optimize the values of the Lagrangian multipliers. For all additional constraints that were relaxed, compared to the basic model, we initially set the Lagrangian multipliers equal to 0. However, more sophisticated initial values may be possible.

In addition, one of the constraints concerning the variation norms was not incorporated. In our approach, we included a variation norm for each depot and an average variation norm over all train drivers. However, we did not consider the fact that there is a high norm value and a low norm value and that we should assign which depots are allowed to satisfy the low norm value only. Nonetheless, we allow for specifying the norm values of the depots individually. Therefore, one can determine which depots are likely to have less opportunities for variation and set the low norm value for these depots and the high norm value for the remaining ones. Furthermore, since penalty costs are included for not satisfying constraints, one could use the high norm value for all depots. In case it is not possible to attain this value for some depots, penalty costs will be charged.

Finally, additional constraints concerning the entire set of duties are present. These constraints were not considered in the given formulations and solution approaches.

8.4 Suggestions for Further Research

As mentioned in Section 8.3, we tested a number of different scoring methods and parameter settings. However, there could be alternative methods or parameter values that are even more suitable. Furthermore, it can be worthwhile to test per instance which method yields best results.

The crew scheduling algorithm applied at NS consists of different subroutines, as described in Section 2.2.1. In this study, the master problem is considered, where duties are selected from a set of available duties, but we assumed the set of duties to be given. However, it could be worthwhile to incorporate the variation restrictions in other phases of the algorithm as well. In the master problem the possibilities to increase variation are limited by the available duties. Therefore, it is recommended to investigate how the column generation phase can be adjusted such that duties are generated that allow for selecting duties with more variation in the master problem.

Finally, as discussed in Section 3.1.1, alternative methods for solving the crew scheduling problem in general have been studied. In this study, we focused on a specific solution approach, since we wanted to extend the algorithm currently used by NS. However, it could be investigated how different methods, e.g. evolutionary algorithms, can be used to solve the crew scheduling problem with variation restrictions and how their performance compares to the presented methods.

References

Abbink, E., 2014. Crew Management in Passenger Rail Transport. PhD Thesis, Erasmus Research Institute of Management, Erasmus University.

Abbink, E., Albino, L., Dollevoet, T., Huisman, D., Roussado, J., Saldanha, R., 2011. Solving large scale crew scheduling problems in practice. Public Transport 3 (2), 149–164.

Abbink, E., Fischetti, M., Kroon, L., Timmer, G., Vromans, M., 2005. Reinventing crew scheduling at Netherlands Railways. Interfaces 35 (5), 393–401.

Cai, X., McKinney, D. C., Lasdon, L. S., 2001. Solving nonlinear water management models using a combined genetic algorithm and linear programming approach. Advances in Water Resources 24 (6), 667–676.

Caprara, A., Fischetti, M., Toth, P., 1999. A heuristic method for the set covering problem. Operations Research 47 (5), 730–743.

Costa, L., Oliveira, P., 2001. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. Computers & Chemical Engineering 25 (2), 257–266.

Dantzig, G. B., Wolfe, P., 1960. Decomposition principle for linear programs. Operations Research 8 (1), 101–111.

Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S., Rioux, B., Solomon, M. M., Soumis, F., 1997. Crew pairing at air france. European Journal of Operational Research 97 (2), 245–259.

Fisher, M. L., 1981. The lagrangian relaxation method for solving integer programming problems. Management Science 27 (1), 1–18.

Hanafi, R., Kozan, E., 2014. A hybrid constructive heuristic and simulated annealing for railway crew scheduling. Computers & Industrial Engineering 70, 11–19.

Huisman, D., 2007. A column generation approach for the rail crew re-scheduling problem. European Journal of Operational Research 180 (1), 163–173.

Jütte, S., Müller, D., Thonemann, U. W., 2017. Optimizing railway crew schedules with fairness preferences. Journal of Scheduling, 1–13.

Jütte, S., Thonemann, U. W., 2012. Divide-and-price: a decomposition algorithm for solving large railway crew scheduling problems. European Journal of Operational Research 219 (2), 214–223.

Kornilakis, H., Stamatopoulos, P., 2002. Crew pairing optimization with genetic algorithms. In: Hellenic Conference on Artificial Intelligence. Springer, pp. 109–120.

Kroon, L., Fischetti, M., 2001. Crew scheduling for Netherlands Railways "Destination: Customer". Computer-Aided Scheduling of Public Transport, 181–201.