Dynamic delay management at railways: a Semi-Markovian Decision approach

Al Ibrahim, A.

Publication date
2010

Chapter 1

General introduction and overview

1.1 Introduction

The Dutch railway system is one of the most heavily utilized railway systems in the world. The railway capacity has become scarce. When delays occur, these delays easily expand throughout the railway network, disrupting it heavily. This demands a careful approach to manage and reduce these delays. Moreover, it is expected that in the future the demand for railway services will raise even further. This holds for both passenger and freight sectors. Building more railway tracks is a very costly solution which in addition takes long time before being implemented, instead, new methods need to be explored to utilize the available capacity in a better way.

As pointed out by Vromans [128], due to interdependencies at the railway network the large part of the delays are knock-on delays which are transmitted from one train onto other. A large part of these delays originates at junctions where trains from different directions meet and intersect each other's paths. If handled correctly, train conflicts at these junctions would have less impact on the network. This would contribute to a more stable network where delay recovery is a fairly fast process and where unrecoverable delays are less damaging.

Many approaches have already been introduced to tackle this train conflict problem. By far most of the approaches are combinatorial of nature. The problem is then formulated as a mixed integer programming model or a Job shop model. One of the major drawbacks of these approaches is that these do not reflect the stochastic, unpredictable, nature of the real world. Most of the operations (train running times, dwell times etc.) are subject to stochastic disturbances. Moreover, when the proposed resolution is not met or an unforeseen event occurs, the whole model must be recalculated. On the other hand the few stochastic models found in the literature are mostly simulation based and lack a number of factors that in our view are important when constructing qualitatively good conflict resolution strategies.

In this dissertation we will examine the possibility of using another technique, based on the theory of the Semi-Markovian Decision processes, to tackle the train conflict problem. We will refer to this approach as the SMD approach. The theory of the Semi-Markovian Decision processes is a well established theory and has a broad range of applications in a variety of stochastic and dynamic systems. The theory has however never been applied in the field of dynamic conflict resolutions at railways before. The focus of the thesis lays more on the methodology of modelling, i.e. how can the railway situation be modelled as a SMD model (i.e. model which is based on the SMD approach) so that it can be used for dynamic conflict resolutions, than on the mathematical technique. Moreover, the emphasis will lay on the situation of a near future where much more trains are expected to run, at least, within the busiest part of The Netherlands and where timetables play a much smaller role than is the case nowadays. The central questions are whether it is possible to model this new railway situation by means of the SMD model and whether the approach gives promising results. And secondly, can the model be applied to a current situation, where timetables play a much larger role, and if so, whether the technique is promising when compared to the conflict resolution method used nowadays by ProRail, the Dutch railway infrastructure manager and traffic controller.

In this first chapter of the thesis we will give the practical motivation for this research and place it in a broader perspective. We will discuss the different performance measures that can be found in practice and we will look at the Dutch railway network. We will then briefly explain the current way of working at the Dutch railways; beginning from the schedule generation until the construction of the conflict resolution rules. Next we will look at the available literature and will briefly explain the theory that our approach is based on. The chapter will be concluded with the outline of the remainder of the thesis.

1.2 Practical motivation

Railways are often associated with timetables which are designed to separate trains from each other in time and space. The idea behind the timetables is that less train conflicts occur and that high-level service towards passengers is ensured. The practice is however different. Train delays do occur and, due to interdependencies within a railway network, result in additional delays to other train services. As a consequence train operations are more stochastic in nature when compared to the initial planning. Furthermore, the demand for railway services (both public and freight) is expected to grow, which requires an increase in capacity. This increase can be achieved either by building more railway tracks or by utilizing the existing capacity in a better way. The former is a very costly as well as short-sighted measure which drives the policy makers towards the second option.

In 2008, the Dutch government expressed its ambitions to intensify the railway transport within the densely populated part of The Netherlands, called Randstad [88]. As part of this plan, the Programme for High-frequency Railway Transport (PHS) [in Dutch: Programma Hoogfrequent Spoorvervoer] has been initiated ([89], [90]). The objective of this programme is to increase significantly the number of train services to facilitate the growth in demand. This growth in demand is found in both the public transportation and the railway freight sectors. Between 2008 and 2020 it is predicted that the passenger demand will increase with 60% to 70% during the rush hours while the transported freight volume is estimated to increase to 100 million tons in 2020 (compare to 28 million tons in 2000 and 45 million tons in 2008) [90]. The PHS programme intends to increase the number of trains within some parts of the Randstad to 6 Intercity trains and 6 regional trains per hour, allowing for a timetable-free operation. The Dutch Ministry of Transport, Public Works and Water Management has announced the project called OV SAAL [87], to be the first project of the PHS programme. The preparations for this project have already started, as announced in the management plan 2009 [99] of ProRail. The project OV SAAL incorporates a track section between vital parts of The Netherlands: Schiphol Airport, Amsterdam, Almere and Lelystad where already in 2012 a high frequency operation is planned. Other track sections will then follow shortly[1]. The pilot study with the code name ETMET, started on the 31^{st} of august 2009 on the corridor Amsterdam - Eindhoven ([33], [127], [92]). The study lasted one week, where in the peak hours, 6

[1]Three more corridors are to be part of the high-frequency operation before 2020: Utrecht - Arnhem/Nijmegen, Utrecht - Den Bosch and the Hague - Rotterdam.

Intercity trains, up to 6 Regional trains and up to 2 freight trains[2] have been running hourly in each direction. The pilot study has shown very promising results ([91]) and an even longer pilot study is scheduled for the last quarter of 2010. For the PHS programme a budget of 200 million Euro has been reserved by the ministry, which is intended to be spent before 2012. An additional investment of 4.5 billion Euro is planned for the period 2012 - 2020 [87]. Above this governmental expenditure, the Dutch Railways (NS) has announced in 2009 to invest 700 to 800 million Euro in the project [94].

The idea behind the high-frequency transport is that the railway operation will be similar to the Metro system where the trains run close to each other. The operation is not entirely timetable-free since the trains are still scheduled to be separated in time with more or less equal intervals in-between. But due to the high frequency one may speak of a timetable-free operation. After all, due to a high number of trains dwelling on stations, no connections between trains need to be defined. This way less delays are transmitted due to train connections. Moreover, the passengers do not need to consider the departure times of trains since the average waiting times are low. Furthermore, it is to be expected that this high frequency operation will lead to less buffer space within the system. Therefore, small delays will more often lead to train conflicts. As a direct consequence, the train arrival times will be more random than is the case nowadays. This change in railway operation will lead to a more dynamic railway service which increases the need for new techniques that can solve train conflicts dynamically.

Even though our primary goal is to design a methodology for future purposes, it is interesting to examine whether the method is applicable for the current situation where timetables are designed but due to delays conflicts arise. In The Netherlands, train dispatchers use the so-called TAD rules (In Dutch: Trein AfhandelingsDocument) to solve conflicts between trains. The TAD rules are constructed off-line and are referenced to whenever a conflict occurs. An example of the TAD rules can be found in Table 8.7 on page 144. Unfortunately these rules are often unsatisfactory. First of all, the TAD rules assume that only one train is delayed at the same time. Subsequently, not all conflict situations are covered by these rules. Secondly, the rules are static and can not handle changing situations. The rules cover only the trains that occur within the timetable. When an 'unknown' train reaches the junction, the train dispatcher is at his own. An example of such an 'unknown' train could be some construction traffic, shunting traffic or an empty locomotive. Further, it is unknown whether the TAD rules are close to optimal.

[2]On the section between Utrecht and Geldermalsen 6 Regional trains were running while on other sections this number was 4. Moreover, 2 freight paths have been reserved. Not all of these paths have been utilized during the pilot.

The rules have never been compared to other rules.

A number of approaches that solve train conflicts, can be found within the literature, however all of them have their drawbacks so that new approaches are a welcomed development.

These factors have inspired us to search for a different approach that could solve train conflicts in the timetable-free environment but which is also applicable for the environment using timetables. In the latter case our goal is to study whether this approach can outperform the TAD rules. We have chosen to describe and model these conflict situations in terms of a Semi-Markovian Decision (SMD) process. The theory of the SMD processes, is well known and is a powerful tool when studying stochastic and dynamic environments. To our knowledge though, this approach has never been used for the purposes of dynamic conflict resolution at railways.

1.3 The research in a broader perspective

As described above, this research examines the possibility of using the Semi-Markovian Decision processes in order to solve railway conflicts. These conflicts can be the result of either a railway system that is timetable free or a railway system with a disrupted timetable.

At railways two kind of delays can be identified. First there are *primary delays* that are a direct consequence of a disturbance. The second category are the *secondary delays* which are knock-on delays caused by delays of earlier trains due to interdependencies within the railway network. Vromans [128] identifies a number of disturbances which can cause primary delays. These are disturbances caused by faulty assumptions in the planning phase (e.g. overestimation of the capacities of rolling stock, usage of longer and heavier trains during operation than has been previously planned, too short dwell times etc.), infrastructure failures (e.g. malfunctioning switches or signals), rolling stock breakdowns (e.g. malfunctioning engine or doors, leaks etc.), human factors (e.g. stochastic nature of driver behaviour), accidents with other traffic or suicides, vandalism, weather conditions.

Due to the unpredictable nature of these disturbances, they are difficult to prevent. There is plenty of research that tries to minimize the probability of such a disturbance to evolve into a delay. For example, studying the boarding behaviour of passengers may help to decrease the probability of the boarding time fluctuations to cause disturbances (e.g. [132], [44]).

Then there are models that aim at cleverly distributing time supplements and buffers within the timetable to reduce the probability of small disturbances evolving into delays. As an example for the work in this direction we refer the reader to [64], [66] and [124]. These models however can not completely prevent delays from occurring. Larger disturbances will still lead to delays.

Vromans [128] states that due to interdependencies at the railway network, the large part of the delays are knock-on delays. Delayed trains run outside of their planned paths[3] and will possibly interfere with the paths of other trains giving rise to train conflicts.

This research aims to resolve train conflicts and in this way to minimize knock-on delays that are the result of such conflicts.

1.4 Performance measures

The performance of the railway companies is judged by various criteria. In The Netherlands, the NS (by far the largest Dutch passenger railway operating company) is judged by the percentage of trains that are more than 3 minutes late[4]. Other countries use other threshold values as the punctuality criterion.

But the punctuality criterion is not the only criterion that can be used to judge the performance of railway companies. An alternative is to look at the percentage of the train connections that are maintained, the total delay of the passenger trains, or the total delay of the railway network as a whole. The above criteria are related to passenger experience, while environmental aspects could also be taken into account. Energy consumption of the trains is a good example of such criterion.

From the infrastructure point of view, other optimisation criteria can be considered. An example for such a criterion is maximizing the throughput through some bottleneck or minimizing the traverse time of the trains within some line segment.

In the research presented in this thesis the conflict resolution strategy will aim at resolving conflicts in such a manner as to optimise one or more of the above criteria.

[3]Scheduled trains are assigned to a certain path within the spacetime continuum. Two separated paths are called conflict-free since the trains that are assigned to them will never be in conflict unless they deviate from their paths.

[4]Starting from the year 2010 the trains are regarded as being late if their delay exceeds 5 minutes [126].

1.5 The railway network and decomposition

A railway network is usually a very complex web containing many hubs and intersections where trains from different directions meet. Figure 1.1 depicts the Dutch railway network.

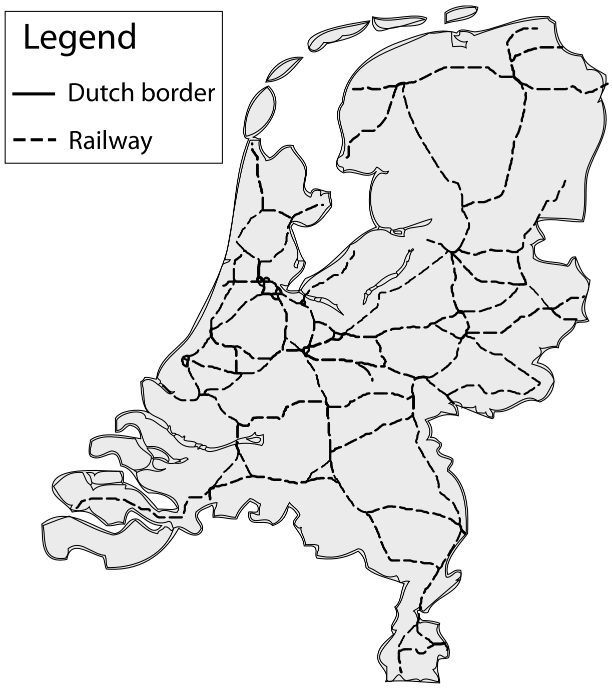

Figure 1.1: The Dutch railway network

When zooming in on a certain hub, an even more complex area will be revealed. Figure 1.2 depicts the situation in the neighbourhood of the Rotterdam Central station. Modelling such an area in an exact way would result in a formulation with an enormous complexity. Such a model tends to become computationally intractable. Instead, an approximation model is needed. A model that captures the essential parts of the conflicting situations and comes up with good dynamic rules for local optimisation. A lot of conflicting situations, regardless of the 'zoom' level essentially come down to the following: trains from different directions come together and compete for some railway segment. In this thesis we will focus on models that describe these situations.

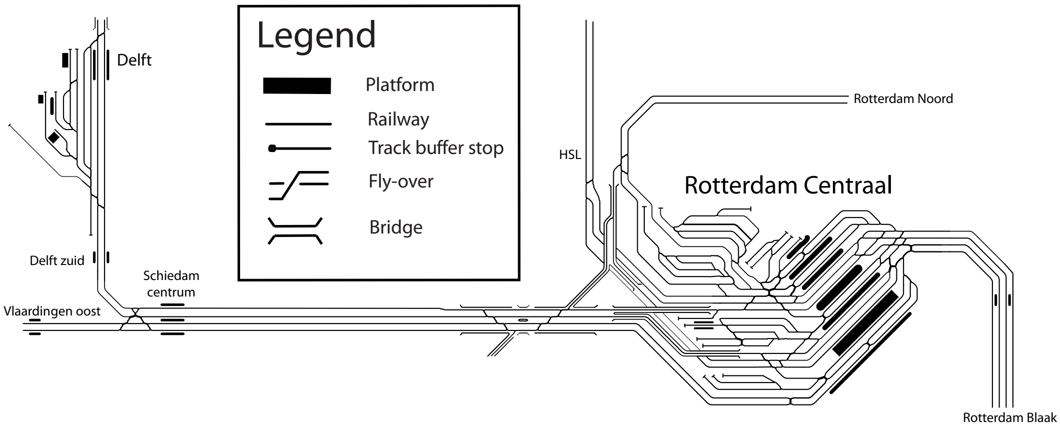

Figure 1.2: The railway in the neighbourhood of the Rotterdam Central station

1.6 Scheduling and Rescheduling in The Netherlands

In this section we give background information about the scheduling and rescheduling as it is being done in The Netherlands. The readers that are familiar with this practice or not directly interested in this subject can skip this section.

The Dutch railway network consists of 2896 km of railway on which 387 stations are located [100]. ProRail is a private organisation that is owned by the Dutch government (the State of The Netherlands is the sole shareholder). ProRail acts as a railway infrastructure manager which is in charge of railway maintenance as well as investments in railway network extensions. It is also a traffic controller in addition to being an independent organisation which allocates the railway capacity by granting concessions to different railway operators. Nederlandse Spoorwegen (NS) is the largest railway operator which shares the available infrastructure with other public transport operators (Arriva, Syntus, Veolia, Connexxion, etc.), a number of cargo operators and some international railway players.

To regulate the railway operations, that together attribute to 145 million kilometres each year [100], a timetable is designed and updated on a regular basis. Since designing a qualitatively good timetable is a very difficult and time consuming process, a decision support system has been built under the name of DONS (Design Of Network Schedules) ([123], [95], [50], [51]). DONS is a timetable design system which helps constructing periodic timetables that satisfy a whole range of different constraints. In The Netherlands, the timetable is periodic, repeating a basic schedule each hour. At rush hours additional trains are inserted into the basic schedule to account for the higher demand.

DONS consists of a database part and a computation part. The DONS database forms the core of the system that contains all the information about the infrastructure (both current and possible future projects), rolling stock and other relevant information. The computation part then uses this information together with some user input to design a timetable. The design is done in two steps. First a global schedule is generated which is done by the CADANS module [107] of DONS. This module presents the timetabling problem as a Periodic Event Scheduling Problem (PESP), first introduced by Serafini and Ukovich in 1989 [112]. The problem is solved by means of a constraint programming based algorithm. The result is a set of arrival and departure times for all considered train services on a global level. The schedule on a local level is constructed by the STATIONS module ([139], [138] and [63]). This module takes the solution of the CADANS module and tries to find a feasible platform assignment and train routing through each station in the network. The result of the calculations is the so-called Basic One-hour Timetable (BOT). Commonly, three BOTs are constructed, one for the off-peak hour, one for the morning peak hour and one for the evening peak hour.

These BOTs provide the basis for the timetables. The next step is to design a Weekly timetable (24x7), which takes the BOTs as basis and alters them by considering traffic fluctuations during the week (less traffic during the weekend and some additional traffic on Monday and Friday due to the weekend). From these Weekly timetables, the Daily timetables are created which consider holidays, special events (football matches, concerts, etc.), scheduled maintenance and other special occasions.

The quality of the timetable produced by DONS is measured by two approaches. The first approach is an analytical model based on the Max-Plus algebra ([46], [31]). The key idea is that the timetable can be represented by a so-called timed event graph which is a Petri net with the element 'time' added to it. The Petri net is suitable since a timetable can be seen as a combination of events which are strongly correlated; an event can not start before one or more preceding events have been completed. The Max-Plus algebra is then used to solve such graphs. This approach has been built into the tool called PETER ([114], [42]). The goal is to measure the timetable robustness by introducing a disturbance and analysing the rate at which the system reverts to the original timetable. This tool is very useful to identify bottlenecks in the system and for forecasting delay propagations through the network. This way, different timetables can be compared. The second approach to measure the quality of the timetables is by means of the simulation technique. The tool SIMONE [85] has been developed to simulate the whole Dutch network and analyse the effect of complicated disturbance scenarios on the whole network as well as the effects of new infrastructure. The tool is also used to compare different timetables. SIMONE is connected to DONS so that different timetables and infrastructures can be simulated. FRISO [86] is another simulation tool which has much more detail of the infrastructure and the train movements than SIMONE and is used to evaluate timetables at a more detailed level.

Another approach to test the quality of the timetable is to analyse real-life realisation data. This approach is different from the two described above, since it evaluates the train operations after these have actually occurred. The tool that is used for this end is called TNV prepare [39] which analyses train punctuality based on train detection data.

After the timetable has been calculated, it can not be made operational before a number of closely related problems are solved. We will mention here the rolling stock circulation problem, the crew planning problem and the shunting problem. For other problems we refer to the overview paper of Huisman et al. [52]. A good paper describing the rolling stock circulation problem, as it is being done by the NS, is written by Fioole et al. [34]. The paper discusses the problem of assignment of rolling stock to the timetable services when both arrival and departure times are known beforehand as well as the expected number of passengers. The problem is formulated as a mixed integer problem and is solved by CPLEX. On the other hand, crew planning is done by TURNI which is based on a mathematical programming technique ([65], [1], [2], [122]). The term shunting refers to the process of parking the rolling stock at a shunt yard. The corresponding planning problem is referred to as a shunting problem. This problem is described by Lentink et al. [73] and Schrijver et al. [108].

Above a very brief overview is given of the techniques that are used when designing timetables in The Netherlands. Unfortunately, during the operation, delays can occur which disturb the scheduled operation of trains and may lead to train conflicts. To resolve these conflicts, the so-called TAD rules are generated. These rules are the result of a negotiation process between different railway operators and ProRail where each operator tries to optimize the situation for its own trains. The set of TAD rules are presented as a guidebook to each dispatcher. Whenever a conflict occurs, the dispatcher looks up the corresponding rule and applies it. More on TAD rules will be discussed in Chapter 8. An interested reader can refer to Table 8.7 on page 144 where an example of these rules is given.

In this section the way of working at the Dutch railways has been discussed. For the tools and approaches used by other railway companies we refer to the technical overview report of Barber et al. [10].

1.7 Literature overview

The dynamic conflict resolution problem, studied in this research, falls within the larger group of problems that carry the name of Railway scheduling problems. These problems have been extensively studied in the literature and are known to be NP-hard [35]. Excellent overviews are given by Assad [9], Cordeau [27], Törnquist [118] and D'Ariano [28]. In her overview paper, Törnquist has classified the relevant literature into three main categories: Tactical scheduling, Operational scheduling and Rescheduling. The three categories have a lot in common but at the same time are very different. The Tactical scheduling is very popular in Europe and involves generation of a timetable long before it is applied in practice. Such a timetable is usually referred to as a Master schedule. The generation of this schedule often takes several months. Thus, the models do not have strict computational time limits. The scope of these models is often very global involving very large networks. The objective is usually to find a feasible timetable that respects a variety of constraints such as meeting the passenger and freight demand, taking into account the availability of the fleet of rolling stock material, personnel and the capacity of the infrastructure. A qualitatively good, conflict-free timetable is preferred here above the computational speed of the underlying algorithms.

Operational scheduling is practised, among others, in North America and Australia and has a much shorter time frame. The schedule is usually generated a couple of hours to days prior to being put to operation. At this level more up-to-date information is available. This way of working suits the accidental nature of operation of the freight railway traffic. The models in this field of scheduling thus have a certain computational time limit but not as strict as is the case with the models of Rescheduling.

While the Tactical and Operational scheduling involved constructing the timetable from scratch, Rescheduling is done when train conflicts arise due to perturbations. In this case, the time for solving the conflict is very limited and the objective of the model most often involves minimisation of delays and/or restoring the initial timetable.

Since the purpose of this thesis is the on-line dynamic conflict resolution, the approaches of direct interest are the approaches describing Rescheduling. However, for the reasons of literature integrity and since the models of both Tactical and Operational scheduling aim at solving problems related to some extend to the Rescheduling problem, we will list these approaches as well. The reader not directly interested in these approaches is advised to continue to Section 1.7.3.

1.7.1 Tactical scheduling

With the tactical scheduling, the emphasis lays more on the quality of the timetable rather than on the computational speed of the underlying algorithm. The quality of the timetable is often measured by its robustness and reliability. The robustness of the timetable refers to its resistance level against stochastic disturbances while the term reliability refers to the level of train punctuality when the timetable is applied.

One of the first pioneers in this field is Szpigel. In his paper [116] he solves train conflicts on a single track line by means of a Mixed Integer program. The line has a number of meets and overtakes where the train order needs to be established. A Branch and Bound algorithm is used to solve the model. Szpigel applies this model to a line section in Brazil with an objective to minimize the weighted travel time of the trains.

The Mixed Integer formulation is adopted by Jovanovic [57] as well. By minimizing the costs of tardiness the dispatching problem is solved where trains are running on a track line consisting of both single and double track segments. In this approach the binary variables are used to indicate the location of meeting point while the arrival and departure times are represented by continuous variables. Later, this approach has been used for the SCAN system by Jovanovic and Harker ([58],[59]). SCAN stands for SCheduled ANalysis and is a decision support system that combines combinatorial optimization and simulation to help decision makers construct robust schedules.

Chiang et al. [24] describe a knowledge-based railway scheduling system which is operational at the Taiwan Railway administration. The master scheduling plan is obtained in two steps. In the first step a global schedule is obtained with an initial train diagram by ignoring train conflicts. These conflicts are then solved by the local scheduler which is advised by an embedded knowledge base.

Carey and Lockwood [22] propose a train dispatching model for a line consisting of a number of stations and links based on a binary mixed integer problem formulation. It supports traffic in one direction only but the speeds of the trains may vary. The model is solved by means of a heuristic approach which first schedules each train individually and then reschedules these initial schedules so that these fit together. The uni-directional traffic assumption is relieved in the follow-up paper [17] where the author shows that the approach is suitable for more general cases.

Brännlund et al. [13] estimate the value of running different types of services at specified times and use these estimates to obtain a schedule that maximizes the 'profit'. A Lagrangian relaxation approach is used to solve the general integer program. The approach relaxes the track capacity constraints. A priority based heuristic is then used to find a feasible solution. Nou [93] extends the paper of Brännlund et al. and presents alternative approaches to generate feasible solutions.

Mackenzie [77] proposes two different approaches to solve the train time-tabling problem for complex networks. The first approach is based on a formulation of a large binary-programming problem which is solved by means of Lagrangian relaxation. The second approach is heuristic in nature, which is based on a probabilistic search procedure. This heuristic generates many alternative timetables then selects one which minimises the total delay. A comparable approach is proposed by Pudney and Wardop [101]. They too, use a probabilistic search technique for generating timetables. By randomly perturbing different data, many hundreds of different train schedules are generated. The algorithm then selects the best schedule based on a total delay cost.

Caprara et al. ([15] and [16]) model the train scheduling problem as a linear integer programming model and use graph theory to represent the problem. The nodes of the graph represent the arrival and departure times. The model is solved by Lagrangian relaxation and is applied to a real-world problem at the Italian railways.

Ingolitto et al. [55] present a constraint programming model and propose an algorithm to reduce the search space. By assigning values to variables and verifying the constraints, different solutions are obtained, each from a different subset of the search space. Then the best one is chosen based on the least average traversal time. The proposed algorithm is shown to be more efficient than the algorithms that search for one single solution in a whole search space. Constraint programming is also used by Salido et al. [106] and Abril et al. [4]. The authors divide the problem into a number of semi-independent sub-problems to reduce the huge number of variables and constraints of the original problem. The performance of sub-models is compared to that of the centralized original model. The authors conclude that the decentralised model is more efficient and has a better behaviour than the latter one.

Zhou and Zhong [136] use a multi-objective train scheduling problem and apply it to a real-world problem where different types of trains with different speeds are running on a double-track railway network. The objective is twofold: to minimize the scheduled waiting times of the high-speed trains and to minimize the total travel time of all trains. A branch and bound technique is then used to solve the model. In their follow-up paper [137] the problem is generalized.

Oliveira [97] maps the single-track railway scheduling problem into a job shop scheduling problem and introduces some real-world constraints by means of the constraint programming. The objective is to minimize the total delay.

Serafini and Ukovich [112] introduce the Periodic Event Scheduling Problem which can be used to construct periodic timetables. Schrijver and Steenbeek [107] introduce an approach that generates feasible periodic timetables for the Dutch railway network. Odijk [95] uses a cutting plane algorithm based on the approach of Schrijver and Steenbeek and applies it to a real-life situation of a modest, but yet non-trivial size. The algorithm turns out to perform very well. A number of papers show how the Periodic Event Scheduling Problem can be extended including the paper of Liebchen and Möhring [76] and Liebchen [75] and the paper of Kroon and Peeters [67].

Goverde et al. [40] introduce the max-plus algebra as a mathematical model to generate periodic timetables. The authors show that essential dynamic characteristics of the system can be quantified including the minimal cycle time and critical circuits. This approach can also be used to predict the propagation of the delays over the railway network. In the subsequent paper (Goverde and Hansen [39]) the authors analyse the Dutch railway operations and introduce the tool TNV-Prepare which filters and prepares train data from log-files for the sake of the detailed statistical analysis. Another model to evaluate timetables stability is proposed by Delorme et al. [32]. The model is based on a formulation of a multi-objective combinatorial optimization set packing problem and is solved by computing a shortest path on a graph. The approach has been tested in France on the Pierrefitte-Gonesse junction and showed that significant gain in robustness of the timetable can be obtained.

1.7.2 Operational scheduling

Operational scheduling is usually done a couple of hours to days from the moment the schedule should be put to operation. As a result, in comparison to Tactical scheduling, more up-to-date information is available and the algorithms have relatively strict time limits.

Kraft [62] develops a dispatching rule which provides departure times, velocities of trains in sections of the line and their stop times at the meeting and passing points. A branch and bound procedure is used to find such a rule with an objective to minimize the weighted sum of delays.

Kraay et al. [61] introduce a mathematical model which generates a schedule that lists the arrival and departure times of the trains together with the speed that these trains should run with to minimize the fuel consumption as well as the train delays. The problem is formulated as a non-linear mixed integer program and solved by means of a branch and bound technique. Another non-linear mixed integer program is proposed by Higgins et al. [48] which incorporates lower and upper bounds on train speeds for each train on each segment. The objective is to minimize the total train tardiness together with the fuel consumption.

Cai et al. [14] describe a heuristic algorithm that can be applied to a single track system and that is in use at a major Asian railway system. The approach generates feasible timetables in a short amount of time. It can incorporate additional constraints and can be used in situations where a new train needs to be added to an existing timetable.

A number of approaches attempt to solve conflicts by doing both: setting the train order and re-routing of trains. Among these approaches is the model of Carey and Carville [18]. The authors address the train platforming problem which assigns trains to time slots and to platforms within complex stations. An algorithm with different search strategies ensures that a feasible schedule is found within a reasonable time. In a parallel paper [19] the authors test the reliability and robustness of the schedule by disturbing the arrival, departure and dwell times in a simulation study. In [20] they extend the scheduling algorithm to a network of complex stations with single or multiple one-way lines in between.

Ghoseiri et al. [36] present a multi-objective optimization model for the passenger train-scheduling problem on a railway network consisting of single and multiple tracks. The objective of the model is lowering the fuel consumption together with the minimization of the total passenger travel time.

Semet and Schoenauer ([110], [111]) seek to reconstruct the original timetable after being perturbed. In doing so, the authors try to minimize the accumulated delays by adapting arrival and departure times and by reallocating the resources (tracks, routing nodes). This is done by an algorithm that is based on a semi-greedy heuristic. The algorithm gradually reconstructs the original timetable by inserting one train after another. By comparing the results of the algorithm to the results of the optimal solution obtained from the integer programming solver (CPLEX), the authors conclude that the approach is promising.

Isaai [56] presents a hybrid approach involving a simulated annealing and a constraint-based heuristic to generate a conflict-free timetable efficiently. First, a fairly good solution is found with the heuristic. Then the algorithm is used to improve the solution quality. This way, the exhaustive search is avoided and the possibility is reduced of being trapped in a local optimum. The approach has been tested with real-life data from the Iranian railway network.

1.7.3 Rescheduling

Very little literature exists on Rescheduling. The issue has been addressed only recently due to the complex nature of the problem and the very limited available computational time. The different approaches that are described in the literature minimize delay propagation by setting the train order at crossing points.

Among these approaches is the model proposed by Adenso-Diaz et al. [5]. The authors describe the on-line conflict resolution problem as a mixed integer programming model and state that solving this problem by means of the Branch and Bound technique is very time consuming. Instead, the authors propose a heuristic approach that intelligently reduces the search space by elimination of certain branches that are considered to be inferior. The approach is implemented at the Spanish national railway company where the tool preselects the best resolution rules and presents them to a train dispatcher.

Törnquist and Persson [120] propose a two level procedure to resolve train conflicts. The relaxation of the mixed integer linear problem is presented where a train that occupies a block is referred to as an event. Ordered vectors link these events to trains and blocks so that a chronological event list is stored at each block and at each train. The upper level of the formulation determines the order of the trains on each block while the lower level allocates start and end times for different events. The lower level is solved by means of a Linear Programming model while two heuristics are presented to solve the upper level. These are Tabu Search and Simulated Annealing. In a subsequent paper [119] the authors extend the approach to the n-tracked networks and examine four strategies to solve the model on a South traffic district in Sweden. They conclude that the choice of time horizon has an obvious effect on problem size, that is, the longer time horizon, the more events and trains are included and thus the more difficult it will be to find a conflict resolution solution in a reasonable time frame. Allowing both re-routing and changing of train order to resolve a conflict turns out to demand too much time. Instead, the authors suggest an approximation strategy, which in most cases does well with respect to computational time and solution quality. This strategy limits the number of train 'swaps' (train order changes with respect to the original order) which speeds up the solution procedure significantly.

Rodriguez [103] presents a constraint programming model for real-time train scheduling at junctions. The model solves train conflicts by means of rerouteing and rescheduling. A branch and bound procedure is used to solve the model. Different constraints bound the search space so that the conflict resolution can take place in real-time. The model has been tested on a problem set taken from a real case study on the Pierrefitte-Gonesse node located in North of Paris. In his subsequent paper [104] the approach is extended to situations were the traffic runs in two directions.

Araya et al. [8] formulate the on-line scheduling problem as a 0-1 mixed integer programming problem which is solved in two steps. First a sub-optimal solution is obtained by a heuristic approach. The branch and bound approach is then used to find the optimal solution. A number of experiments show the efficiency of the approach in terms of computational time.

Another approach is to formulate the train conflict problem as a Job-Shop problem. Here, the trains are jobs and the tracks are machines. The problem is then to find the best assignment of the trains to the tracks so that the overall delay (or some other optimization function) is minimized. Mascis and Pacciarelli [78] introduce blocking and no-wait constraints to the Job-Shop scheduling problem and use an 'Alternative graph' to solve it. Within the Alternative graph the nodes represent operations (the occupation of a certain train on a certain block section) and the arcs represent relations between the nodes. A relation can be either of type *precedence relation* indicating that an operation can not start before the other or of type *alternative relation*. The latter consists of two conflicting arcs. A solution is then feasible when only one arc is chosen from each alternative relation. Blocking and no-wait constraints fit perfectly in the railway scheduling. With a blocking constraint a job, that has completed its processing time on a machine, remains on it until the next machine becomes available for processing. Thus the train is blocked when the next track is occupied. Moreover, by introducing the no-wait constraint, no time is lost between two subsequent operations, thus, when a train leaves one track, it simultaneously enters the next one. Introducing these two constraints makes the problem computationally more difficult than the case with unlimited buffers. The authors show that the existing solution procedures are unlikely to produce feasible and qualitatively good solutions for the blocking and no-wait job-shop problem. D'Ariano et al. [30] use this idea to solve train conflicts within the perturbed timetable by minimizing the deviation from the original timetable. The authors develop a truncated version of the branch and bound algorithm to find near optimal solutions within short time limits. The approach has been tested on a heavily congested area of the Dutch railway network and showed promising results. The approach has been implemented in the decision support system ROMA [29].

Alternatively, the ideas of Mascis and Pacciarelli are adapted within the COMBINE project ([98], [38]) and COMBINE 2 project [37]. The two projects aim at automation of traffic management and have lead to the development of the traffic management system (TMS) that resolves train conflicts in real-time. This system [80] consists of two modules. The first is called the Conflict Detection and Resolution module where the conflicts are solved by means of the alternative graph. The second module (Speed Profile Generator) takes the order of trains that are the result of the first module and generates feasible speed profiles which are optimized with respect to energy consumption.

Yet another approach has been given by Ho et al. [49]. By means of dynamic programming a complex problem of finding the best order of trains to move through the junction is decomposed into smaller sub-problems. When a conflict occurs, the algorithm is called and the trains in the conflict area are taken into account. Only i trains from the n trains that are found within the area are being considered. The idea behind this, is that considering all n trains is regarded to be redundant. Given the arrival distribution function, a new arrival is expected to take place after i trains have managed to cross the junction. New arrivals change the situation and make previous conflict resolution sub-optimal.

Vernazza and Zunino [125] propose a decentralized approach to resolve train conflicts within a network. This way, the approach overcomes the difficulties found in the centralized approaches with respect to limited computational times. The approach is based on a resource allocation technique where priority rules are applied to local control decisions.

The idea of distributed decision making has also been analysed by Lee and Gosh [72]. The authors describe a decentralized algorithm called RYNSORD and test its stability through different perturbation scenarios. The algorithm appears to be strongly stable with respect to perturbations on input traffic rate but unstable when permanent failures occur on track segment and communication links.

Lamma et al. [68] propose a decentralized approach where local modules produce dispatcher rules that obey a set of constraints. Some techniques are used to reduce the search space of the sub-problems. The prototype model is tested on small and average-sized stations and show a potential of the approach. The subsequent paper [69] enhances the model by removing some limitations.

Chiu et al. ([26], [25]) formulate the train rescheduling problem as a constraint satisfaction problem. The train rescheduling algorithm is based on a constraint propagation approach which minimizes both the passenger delay and the number station visit modifications. In order to make the approach applicable for real-time purposes two heuristics are presented to speed up the solution procedure. The authors verify the feasibility of the approach by testing it on a real-life data.

Above a literature overview of the deterministic rescheduling models is presented. The presented models formulate the problem as Mixed integer programming models, Job-Shop problems, a Dynamic programming formulation, some decentralized approaches and constraint satisfaction problems. The models require large computational times so that heuristic methods are used to obtain solutions within reasonable time frames. These heuristics however do not guarantee the optimality of the solution. Moreover the solution is only feasible within the current context. If the situation changes or the proposed schedule is not met the whole model needs to be recalculated. In the next section stochastic approaches are discussed. These approaches attempt to model uncertainties which are found in the real world (think of the running times, dwell times and other operations which are often stochastic).

1.7.4 Stochastic approaches

Little literature describes stochastic approaches for railway scheduling even though the incorporation of stochastic elements can lead to more robust schedules. In the real life, the running times and dwell times are often subject to perturbations. Moreover, a temporary unavailability of some resource due to maintenance or failure often leads to additional delays.

The very few stochastic models that can be found in the literature are designed to enhance the quality of the timetable while virtually no stochastic models can be found that aim to solve train conflicts in real time. A good overview of the first type of stochastic models is done by Hansen [43]. The author states that the timetables are based on deterministic running, dwell and headway times and that the amount of time supplements and margins found in the timetable is mainly based on rules of thumb and only seldomly derived from statistical analysis. On the other hand, from an empirical analysis conducted on train detection data at different Dutch railway stations it follows that the mean speed of trains approaching a station is 10 to 20 % lower than the designed speed. Moreover, the scheduled dwell times are also often exceeded. Absence of these stochastic elements undermines the quality of the timetable. Two types of delays can be identified: the scheduled waiting times and the non-scheduled waiting times. The first results from the difference between the assumptions made during the planning and the actual conditions during operation (e.g. overestimation of the possibilities of train characteristics or are the result of rounding the arrival and departure times to meet the timetable). On the other hand, the non-scheduled waiting times emerge from unforeseen events during operation. The scheduled waiting times can be used as an indicator for the quality of the timetable. In 1974, Schwanhäusser [109] uses the mean queue length as an estimation of the quality of a timetable and develops a stochastic approach based on a M/D/1 model to estimate the mean length of the queue. Later, Wakob [129] and Hertel [47] enhanced this model by relaxing the assumption that the arrivals follow a Poisson process.

Wendler [130] extends the model to a three-train model of type G/G/1 and uses this model to estimate the available time lags between the headway times. The obtained model is used to model independent random requests for infrastructure capacity that are placed by different train operators and measure the scheduled waiting time that is the result of the different train assignments. In another paper [131], Wendler presents an approach predicting the scheduled waiting time by means of a $M/SM/1/\infty$ queuing model and uses this model to predict the quality measure for different bottlenecks. The Semi-Markovian service times arise from the fact that different train types require different headway times (the minimal safety time required between two subsequent trains) while the infinite waiting area implies that trains cannot be lost and will be served even when heavily delayed.

Analysis of the total waiting times is also done by means of simulation tools. Among these tools are RailSys [10], OpenTrack [10], SIMONE [85], STRESI and FRISO [86] where the effect of primary delays can be studied within a detailed railway environment. The first three tools are macro simulation tools which model huge networks. Different type of delays are introduced by means of dwell time, arrival and departure time disturbances. The conflicts are solved either by FCFS principle or by means of deterministic rules. FRISO is a micro-simulation tool which has a very high detail of railway infrastructure, signalling system and train dynamics. As a conflict resolution mechanism, FRISO uses FCFS, deterministic rules or can be connected to some external conflict resolution system. Another tool called ANKE is developed at RWTH Aachen [121] which makes estimations for the scheduled and unscheduled waiting times.

A probabilistic model is given by Yuan and Hansen [135] which models running times, arrival and departure times as well as dwell times as stochastic variables. The Stieltjes convolution of individual independent distributions is then used to solve the model. The model has been validated by comparing the different values to the track occupation and release data recorded at The Hague HS station.

Another approach is based on a Periodic Event Scheduling Problem which has been developed by Vromans [128]. The author introduces exogenous disturbances and monitors the propagation of the resulting delay. The linear programming model then minimises the mean arrival delay over all runs by optimally allocating time supplements along the line. The approach is applied to the line section Haarlem-Maastricht where the author states that the punctuality can be risen significantly by allocating time supplements differently.

Huisman and Boucherie [53] present a queueing model that captures both scheduled and unscheduled train movements and investigates the delays that emerge from the fast trains catching up with the slow ones on a single track line. A system of linear differential equations is used to obtain running time distributions for each train service. In another paper, Huisman et al. [54] look at the situation where the timetables are not yet known and propose a solvable queueing network model to obtain closed form expressions for mean delays. This way, new network designs and capacity expansions can be evaluated for which no timetables exist yet.

Carey and Kwiecinski [21] develop a simple stochastic model to approximate the knock-on delays that arise on n-track lines as a result of fast trains catching up with the slower ones. The model is tested by a stochastic simulation where interactions between trains are modelled. The authors state that the found relation can be used to enhance train planning models and increase the quality of the timetables that often ignore knock-on delays.

Another model estimating the knock-on delays is proposed by Yuan [133]. The analytical probability model predicts the train punctuality at stations by taking into account dwell times, stochastic interdependencies between train movements, speed fluctuations and the dynamic delay propagation.

Stochastic approaches for the purposes of the real-time Rescheduling are rare. Among these, is the approach proposed by Sahin [105]. Sahin has analysed dispatchers' decision processes while resolving a train conflict and has identified four factors that the dispatchers consider prior to deciding which conflicting train to stop. He then constructed a heuristic algorithm for a single-track line that detects the trains involved in the conflict and applies a look-ahead method to find out the best resolution. The method stops each train sequentially and calculates the consequences. At scheduled points, the expected arrival time of every train is calculated by considering their potential conflicts and associated average delays. The algorithm then selects the train order which minimizes the sum of the deviation from the expected arrival times. Furthermore, the author presents a multi-attribute choice algorithm that should help dispatchers determine the dynamic priorities.

Cheng [23] models the train traffic rescheduling problem as a resource-constrained problem. The focus of the paper is solely the crossover conflict situations where trains cross each others paths only for a short amount of time and continue their movement in different directions. The paper discusses an innovative simulation approach which is faster in detecting conflict situations than is the case with plain event-driven simulation. The approach, which the author refers to as a Hybrid simulation approach is a combination of the network-based simulation and the event-driven simulation. Train traffic behaviour is expressed by the weighted directed network where arrival and departure events are represented as nodes and different constraints as arcs. Resource conflicts are represented by disjunctive arcs, i.e. two nodes connected with two arcs pointing in different direction. The presented algorithm solves the conflicts by selecting one of these arcs so that the total train delay is minimized.

A different approach is proposed by Medanic and Dorfman ([81],[82],[83]). The authors use dynamic systems to dynamically reschedule trains. The problem is formulated as a discrete-event dynamic system where the events are the arrival times of trains at the so-called meet and pass nodes where trains can overtake each other. The conflicts are solved in two stages: First in train optimization stage, one determines optimal pacing velocities for each train in each section with respect to energy consumption. Then in the scheduling stage, the average pacing velocities computed in the first stage are used to find the best train order. This decomposition method preserves the minimal energy costs while determining a feasible time-efficient schedule. The train order is determined by a set of rules which give the train the right of way if it is the first one to reach the next meet and pass node. The same holds for the bi-directional situations where the conflicting trains are located at two different nodes and want to run towards each other. In this case, the fastest train will get the right of way. In their subsequent paper [84], the authors extend the model by introducing double tracks to the network and allowing for train priorities. A conflict involving a priority train will be solved by giving this train the right of way. The authors state that the presented approach is less sensible to perturbations than the approaches based on mixed integer programming where the whole solution needs to be recalculated when the trains do not keep up with the previously recalculated schedule. Li et al. [74] have improved the model of Medanic and Dofrman by introducing an improved simulation method and making use of more global information. This global information should lead to less conflicts later on. The idea is that given the current position of all the trains moving on the route of the train, the current conflict should be resolved as to minimize the possibility of future conflicts.

1.7.5 Other related models

Railway scheduling problems have many similarities to problems encountered in airline scheduling (Wendler [131]). Wendler observes that the time needed for trains to cross a junction can be modelled as semi-Markovian service times. The term 'semi' refers here to the fact that slow trains need more time to cross the junction than fast trains. The same principle can be found in airlines when modelling the landing of an aircraft. Due to turbulence, the landing aircrafts must be separated in time. The size of the aircraft influences the magnitude of the turbulence and thus the minimum separation times.

Excelent overviews of approaches addressing airline scheduling problems can be found in papers of Odoni [96] and Beasley et al. [11]. Other interesting contributions are made by [113],[70],[12],[79],[115],[3] and [60].

1.7.6 Final remarks

Above an extensive literature study has been given. Both off-line scheduling (Tactical and Operational scheduling) and on-line scheduling (i.e. Rescheduling) models have been discussed. Most of the models are deterministic and only a few are stochastic in nature. The deterministic models do not account for the uncertainties that are present in reality and treat various operations (e.g. running times, dwell times etc.) as being deterministic. Goverde et al. [41] has analysed the realization data at the Eindhoven station in The Netherlands and concluded that the time prolongations of different railway processes fit well to exponential distributions. The same conclusion is drawn by Yuan [134] who analysed another Dutch station: The Hague HS. Moreover, the deterministic models make use of the rolling horizon and do not take into account the trains that fall outside the scope even though current decisions may have direct impact on these trains (e.g. stopping a train in favour of traffic from another direction has consequences to trains behind it). Another drawback of deterministic models is that the conflict is resolved by generating a feasible schedule. When however due to some perturbation or other unforeseen event the schedule can not be met, the whole schedule should be recalculated.

On the other hand, stochastic models that are found in the literature are mostly meant for improvement of timetable quality and not for dynamic Rescheduling. The very few stochastic Rescheduling models have only limited optimization capacities. The resolution methods are often a set of basic rules which consider only a few factors that influence the optimal strategy. One of the factors that we miss in all of the approaches is the fact that the conflict resolution has a direct consequence to the trains that are running behind the conflicting trains and will be affected indirectly by the decisions taken. In dense networks as the network of The Netherlands, stopping a train on a heavily used track will have a huge impact on the situation later on. In our view, the usage intensity of the tracks is an underestimated factor and we therefore want to look into new techniques where such a factor is an integrated and implicit part of the model.

In this thesis we examine the possibility of using the technique of semi-Markovian decision processes for dynamic conflict resolution. In this approach the stochastic elements are combined with the dynamic programming idea of solving a complex problem in stages. The SMD-algorithm takes future arrivals into account by means of stochastic arrival information. Moreover, the trains that have already passed the conflict site are included in the state space. These trains influence the future decisions in the sense that when a slow train is directly behind the conflict site, sending a fast train first has only a limited advantage. To our knowledge, this approach has never been used for railway conflicts before.

We will show that this approach is powerful enough to incorporate important elements that shape up the optimal decision strategy while keeping the model small enough for practical applications.

1.8 The Semi-Markovian Decision technique

As stated, the purpose of this thesis is to examine the possibility of using the so-called Semi-Markovian Decision-technique (in short SMD) to tackle train conflict problems. In this section we will explain the SMD-technique, outline its basics and present some terminology that will be used in later chapters. For a more extensive discussion of this theory we refer to text books such as Puterman [102] and Tijms [117].

Semi-Markovian Decision processes are used to analyse dynamic systems, where present decisions can affect future situations and future decisions. The situation at railways is exactly that. The system is constantly in motion and since the trains can overtake each other at a limited number of places only, the train order, which is the result of the decisions made at a certain point in time, affects the system for a long time.

1.8.1 Introduction to SMD

The Semi-Markovian Decision process is a mathematical technique which is suitable for situations where decisions need to be taken within a dynamic system. The decision (or action) to be taken depends on a situation (i.e. state) the system is currently in. When a decision is taken, the system changes from one state into another. The time needed for this change is stochastic and depends on the current state and the selected action. During this time, a number of uncertain events can take place resulting in a range of possible new states, each occurring with a certain probability. Thus, the system evolves from one state into another as a result of the selected action and the sequence of stochastic events.

The time between two decisions will be called a slot. Each decision involves (expected) direct costs which depend on the decision, the state at the start of the slot and on the length of the slot (sometimes, the models are formulated in terms of rewards instead of the costs which in essence are negative costs). The objective of the model is to minimize the costs over a finite horizon or in case of an infinite horizon, minimize the average costs per time unit. An alternative model, not discussed here, is to discount future costs and to minimize the total expected discounted costs.

The SMD model description requires the specification of the following components: States, decisions, transitions and costs. In what follows we will discuss each component in more detail.

States Markovian systems are characterized by being memoryless. That is, having knowledge about previously visited states does not provide additional information and does not change the way the system evolves from state to state, nor does it influence the process of decision making. Thus, it is enough to describe the state by the current situation only.

In the problem description that we will be using in this thesis, the natural state description will be a vector. Let variables x_1, x_2 up to x_n together define the current state then the state of the system is represented by the vector $\mathbf{x} = (x_1, \cdots, x_n)$ where each $x_i \in \mathcal{X}_i$.

All states together form the *state space* which in this case is defined by $\mathcal{X} = \prod_{i=1}^{n} \mathcal{X}_i$. It is an infinitely countable space which is often truncated to obtain a finite one for computational purposes.

Decisions Each time slot a decision a is taken. This decision takes place at the beginning of the slot. An alternative, which will not be discussed here, is to take decisions at the end of the slot. Let $\mathcal{A}$ be the set of possible decisions then $a \in \mathcal{A}$. States may have a constrained decision space, leading to a state dependent decision space $\mathcal{A}(\mathbf{x})$ where $\mathcal{A}(\mathbf{x}) \subset \mathcal{A}$. Some model settings allow for multiple decisions to be taken simultaneously. In such systems, the decisions will be multi-dimensional.

Transitions Each decision takes some time to complete. The transition time depends on the state at the beginning of the slot and on the selected decision and has a mean $\tau(\mathbf{x}, a)$. During this time, one or more (stochastic) events can occur which result in a number of possible future states $\mathbf{x}'$, each with probability $p^a_{\mathbf{x}\mathbf{x}'}$.

Direct costs If decision a is taken in state $\mathbf{x}$ and results in new state $\mathbf{x}'$, then direct costs $c(\mathbf{x}, a, \mathbf{x}')$ are incurred per time unit. The total costs of such a transition is thus equal to $c(\mathbf{x}, a, \mathbf{x}')$ multiplied by the length of transition time $\tau(\mathbf{x}, a)$. As state $\mathbf{x}$ can lead to different states $\mathbf{x}'$, usually the expected direct costs $C(\mathbf{x}, a)$ are computed. These expected costs depend on the state at the beginning of the slot $\mathbf{x}$, the selected action a and on the transition probabilities $p^a_{\mathbf{x}\mathbf{x}'}$:

$$C(\mathbf{x}, a) = \sum_{\mathbf{x}'} c(\mathbf{x}, a, \mathbf{x}') \cdot \tau(\mathbf{x}, a) \cdot p^a_{\mathbf{x}\mathbf{x}'}$$

1.8.2 Solving the SMD model

To solve the Semi-Markovian decision model, one needs to transform it to a standard Markovian Decision Problem (MDP) for which good numerical solution techniques exist. The difference between the two is that the latter model has fixed slot lengths. As a result both costs and transition probabilities are no longer slot length dependent.

To transform the model, define the minimal slot length variable τ to be $\min_{\mathbf{x},a} \tau(\mathbf{x}, a)$ and resize the costs and transition probabilities by multiplying both by the factor $\frac{\tau}{\tau(\mathbf{x},a)}$. Since τ is equal to the minimal value of $\tau(\mathbf{x}, a)$, many of the original transitions require more than τ time units. For this reason, some artificial transition probability mass is added for the transition from state $\mathbf{x}$ to $\mathbf{x}$ equal to $p^a_{\mathbf{xx}} = 1 - \frac{\tau}{\tau(\mathbf{x},a)}$.

Now, the MDP can be solved by one of the three solution techniques: Value iteration (or Successive approximation), Policy iteration and Linear Programming. In this thesis we will use the Value iteration technique to solve the models. The choice for this solution technique has been affected by the fact that the computational complexity of the Value iteration algorithm is $\mathcal{O}(|\mathcal{A}| \cdot |\mathcal{X}|^2)$ while that of the policy iteration is $\mathcal{O}(|\mathcal{X}|^3)$. In the models that we will be constructing, the $|\mathcal{A}|$ is very limited in comparison to the $|\mathcal{X}|$. Moreover, the lack of convincing results of the linear programming technique, found in the literature, has drove our decision towards the Value iteration technique. Thus in the remainder of this section we will confine ourselves to the discussion about the Value iteration technique. Moreover, we will discuss here only the aperiodic case with infinite horizon. For the reader interested in periodic solutions, finite horizons, Policy iteration or Linear Programming solution techniques we refer to Tijms [117] and Puterman [102].

Let $V_n(\mathbf{x})$ be the value function which represents the total expected costs over the n slots when starting in state $\mathbf{x}$. The Value iteration algorithm runs as follows:

Value iteration algorithm

Step 1 Set $n = 0$; $V_0(\mathbf{x}) = 0 \ \ \forall \mathbf{x} \in \mathcal{X}$ and
define ϵ to be a very small number compared to $C(\mathbf{x}, a)$

Step 2 $n = n + 1$; compute for all $\forall \mathbf{x} \in \mathcal{X}$:

$$V_{n+1}(\mathbf{x}) = \ min_{a \in \mathcal{A}(\mathbf{x})} \Bigg(\frac{\tau}{\tau(\mathbf{x}, a)} C(\mathbf{x}, a) + \frac{\tau}{\tau(\mathbf{x}, a)} \sum_{\mathbf{x}'} p(\mathbf{x}, a, \mathbf{x}') V_n(\mathbf{x}') + (1 - \frac{\tau}{\tau(\mathbf{x}, a)}) V_n(\mathbf{x}) \Bigg) \quad (1.8.1)$$

and store a minimizing action a where $\pi(\mathbf{x}) = a$

Step 3 Compute bounds:
$U_{n+1} = max_{\mathbf{x}}[V_{n+1}(\mathbf{x}) - V_n(\mathbf{x})]$ and
$L_{n+1} = min_{\mathbf{x}}[V_{n+1}(\mathbf{x}) - V_n(\mathbf{x})]$

Step 4 stop if $span(V_{n+1} - V_n) = U_{n+1} - L_{n+1} < \epsilon$
else go back to step 2.

The algorithm starts by setting $n = 0$ and $V_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$. Then by incrementing n, the recursive function (1.8.2) is computed for each $\mathbf{x} \in \mathcal{X}$. The minimizing action a is then stored in vector $\pi(\mathbf{x}) = a$ for each state $\mathbf{x}$. The algorithm stops if the $span(V_{n+1} - V_n)$ is smaller than some pre-specified small number ϵ.

When the algorithm stops then π is a nearly optimal stationary policy, i.e. the one that minimizes the long-run average costs g^π which are approximated by $\frac{L_{n+1}+U_{n+1}}{2}$ and are ϵ close to the optimal average costs g^*, i.e. $g^\pi - g^* < \epsilon$.

Each iteration of the value iteration algorithm has a computational complexity of $\mathcal{O}(|\mathcal{A}| \cdot |\mathcal{X}|^2)$. This is due to the fact that at each step $|\mathcal{X}|$ states are considered where a maximum of $|\mathcal{A}|$ decisions are evaluated. Each state-action pair $(\mathbf{x}, a)$ may lead to at most $|\mathcal{X}|$ future states.

1.8.3 SMD as an application for railways

In our research on dynamic delay management at railways, we translate the railway situation into the SMD model. The model we use, focuses on dynamic conflict resolution and thus will try to resolve the conflicts and optimize the situation with respect to some kind of objective function.

A typical conflict situation can be described as a junction consisting of two parts. The first part is a set of tracks where the trains arrive at. These tracks will be referred to as *arrival tracks*. The second part consists of the track that the trains move to after crossing the junction and will be referred to as *destination track*. Throughout the thesis we will examine the models where the second part consists of exactly one single track. We will show that this model can already be applied to a variety of cases. In the last chapter we will discuss how the model can be extended to cases where the second part consists of more tracks.

Each track of the junction has a certain length and is divided into smaller blocks where only one train can run at the same time. We will refer to a junction as being a system. The state of the system will be described by the position of the trains and their types. Also their speed and direction of movement is part of the description. The decision is then fairy simple and will give the right of way to a train from one of the arrival tracks.

By far the most challenging part is the modelling of transitions. This part is not straightforward and various choices need to be made. After the decision has taken place, a number of changes occur. First of all, if the decision gives some train the right to cross the junction, then the train will need some time to approach the junction, cross it and clear it for the following train. In the meantime, the junction is blocked for the rest of the traffic. This time period is the slot time, mentioned in the previous section. At the end of this slot time, the trains on the destination track have changed their position, the train with permission has crossed the junction, the speeds of the trains on the rest of the arrival tracks are likely to be affected (due to the blocked junction) and new arrivals have entered the arrival tracks. Trains on the destination track run with different speeds but can not overtake each other. Therefore their speed depends on the speed of the preceding trains and the distance towards them. Moreover, there is a minimal safety interval that the trains should obey. All these factors will lead to a fairly complex transition structure which depends on the state at the beginning of the slot, the selected decision and on the length of the slot.

Another modelling aspect concerns the arrival process of trains. The primary goal of the model is to capture railway systems where no timetables are used. Such systems are likely to be implemented in the future. The SMD model will thus not hold any information about timetables and will assume train arrivals to follow a Poisson process. This assumption is a simplification of the railway situation but does approximate it to some extent. Moreover, the approximation tends to fit better when the number of trains increases. Also the model allows for different arrival processes that can be modelled as a Phase type process which is a less chaotic arrival process and may be a better approximation than the Poisson process.

While the assumption that the arrivals are chaotic does not hold for the present day situation where timetables are a common practice, we would still like to test the approach within such an environment and compare its performance to the TAD rules which are used by ProRail nowadays. Due to the delays within the railway network, the arrivals are stochastic to some extent. Moreover, since our model captures a number of important factors which influence a dynamic conflict resolution strategy we expect the model to perform quite well in practice.

1.9 Main goals

Throughout the previous sections the goals of the thesis have already been mentioned. In this section we would like to summarize them. The main goal of the thesis is to examine the possibility of using the Semi-Markovian decision technique for dynamic conflict resolutions at railways. As pointed out by Vromans [128], due to interdependencies at the railway network the large part of delays are knock-on delays which are transmitted from one train onto other. This occurs primarily at the junctions and at the track sections behind these junctions when a fast train catches up with a slower one. The goal of our research is to optimize the situation at junctions and taking into consideration the tracks behind them.

Currently the timetables are widely used in the railway world. Unfortunately these timetables are not always met which sometimes leads to perturbed timetables. In addition, there is a tendency, at least in The Netherlands, to increase the number of trains substantially and move towards a timetable free environment. The idea of our research is to construct a model that is applicable for both timetable and timetable-free environments.

The second goal is to examine whether the technique is promising when compared to the so-called TAD rules, the conflict resolution method used by ProRail nowadays. For this, we will look whether the SMD rules yield better results in terms of train punctuality than the TAD rules and whether the rules themselves are as clear and easy to comprehend as the TAD rules. The latter is important as the rules are applied by train dispatchers that need to understand the rules they use.

1.10 Thesis outline

In this chapter we have introduced the main goals of the thesis and have placed the research in a broader context. We gave the practical motivation and briefly explained the current way of working at Dutch railways. A literature study has been conducted and the standard Semi-Markovian Decision (SMD) theory explained.

The purpose of the next chapter (Chapter 2) will be to address a number of model preliminaries. The goal of the thesis is to optimize the situation around junctions. Thus in Chapter 2 we will be talking about the elements which are common to different type of junctions and will discuss the ways in which these elements can be modelled. A number of modelling choices will be explained and some concepts, which are used throughout the thesis, will be outlined.

In Chapter 3 we will show in detail how a simple junction can be modelled with the SMD technique. At this type of junction, trains from different directions meet and share the same infrastructure from that point onwards for some time. It then needs to be decided which train to give the right of way. This type of junctions is the most common at the railways. Each element of the model (states, decisions, costs etc.) will be thoroughly explained. Special attention will be given to the element Transitions, which is by far the most challenging part to model. Also the state space reduction technique and the model complexity will be reviewed.

In Chapter 4 the model of the previous chapter will be solved for different scenarios in order to study the structure of the solution and examine its performance. The solution of the model is a conflict resolution strategy, which we will call the SMD strategy. By means of the simulation technique the performance of the SMD strategy is compared to that of a number of other strategies. We will examine the differences and explain why the SMD strategy performs well.

Then in Chapter 5 we will show how the model can be extended by introducing bidirectional traffic. The major difference arises from the fact that some track segments, when occupied, are blocked for the traffic coming from the opposite direction. We will show that this requires only slight changes to the existing model and will then compare the performance of the bidirectional model to that of other strategies.

The models addressed earlier are intuitive but have a drawback of having a large state space. A smaller state space leads to compact models which are easier to compute. As a result, more complex situations can be analysed. Another reason for looking for model enhancements is that the model of the previous chapters does not fully support the so-called Headway concept which states that the trains need to be separated in time by a certain interval. The model of Chapters 3 - 5 respects this concept at the beginning and the end of each track but can violate it halfway through the tracks. In Chapter 6 we will therefore introduce a more compact way of modelling to correct the inefficiency of the earlier model. Furthermore, the new model will always be consistent with the Headway concept.

In the previous chapters only isolated junctions were examined. As the railway network is a collection of junctions where interdependencies arise, we will examine in Chapter 7 different network settings and we will show how the network can be divided into a number of sub-areas. For this, the concept of the junction scope will be introduced and we will show how the SMD model can be constructed for each of the sub-areas. By means of simulation, the performance of the local SMD strategies is studied and the results viewed at the level of the network. The performance of the strategy is compared to that of other strategies.

Up to this point, only theoretical junctions have been considered. In Chapter 8, we shift our attention to a real-life situation. The first aim of the chapter is to examine whether the complexity of the real-life situation can be modelled with the SMD approach. The second purpose is to examine whether the SMD approach, which has been primarily designed for a timetable-free situation, performs well within the timetable environment. In cooperation with ProRail a test case has been selected which involves a complex line segment where delays frequently arise and innovative techniques are needed to handle train conflicts. In this chapter we will show how to tailor the SMD-approach to model these complex railway hubs and we will compare the results of the decisions of our approach to those of other strategies. A strategy of a particular interest is the so-called TAD-strategy. This is the conflict resolution strategy which is currently being used by ProRail. Thus the questions to be answered are whether the SMD strategy can be used within a situation where timetables are used and whether the model can be used as an effective substitute of the TAD strategy.

The dissertation will be concluded with an epilogue, highlighting the possibilities and the limitations of our approach and addressing some possible future research areas.