Preface

In this thesis the algorithmic foundations of *train classification* are studied. Train classification is a special sorting problem originating from the field of railways. It refers to the process of splitting up freight trains into their single cars and reassembling them to form new trains, a process which is essential for freight transportation. The train classification process takes place in special installations of railway tracks and switches called *classification yards*. There are *inbound trains* arriving at the yard that carry the cars which are fed in the classification process. The sorting process is applied following instructions called a *classification schedule* and uses the yard's track facilities to arrange the cars in different compositions and orders. Connected to engines, these compositions eventually leave the classification yard as *outbound trains*.

Call for Improvement

The typical freight car is reported to spend an average of 62 % of its lifetime in classification yards and only 6.6 % in service [Kum04], which calls for fundamental optimization of the classification process. In their recent article [HL10], Heydenreich and Lahrmann see significant opportunities to improve the performance of wagonload traffic. As the key points to improve efficiency and profitability, they particularly name a higher utilization of freight cars and shunting engines as well as an increased efficiency of classification yards. Also, in an interview with the Executive Vice President and Chief Operating Officer of *Canadian Railways* [Har00], reducing the dwell time of cars in railway yards to increase their utilization is identified as one of the key factors to improve railway service.

These dwell times may be reduced by accelerating the classification process, which can be achieved by minimizing the amount of movement of cars and shunting engines. It would be too expensive to extend or redesign classification yards that were designed decades ago to accommodate traffic requirements substantially different from today. An obvious way to increase the performance of existing yards is to improve the train classification process itself. The methods applied today are mostly designed from experience and may yield practically acceptable solutions. However, they offer a lot of potential for optimization. To this aim, the underlying combinatorial structure of the practical problem must be understood in order to provide efficient algorithmic solutions, to prove their solution quality, and to show that these solutions work in practice.

In this thesis a suitable theoretical model for the train classification problem is provided, which is applied to develop algorithmic solutions. The rich variety of variants of abstract problems, resulting from numerous constraints occurring in the multifaceted practical problem, as well as the arising robustness questions yield an interesting field to investigate novel algorithmic methods. Furthermore, reversely applying the gained insights to real-world railways provides a fundamental step towards a train classification process significantly more efficient than today's methods.

Summarizing, train classification presents both an interesting theoretical field of computer science research and, as it often is the bottleneck in freight transportation, an important practical problem with a high potential for improving railway freight transport.

Thesis Outline and Summary of Results

The reader is introduced to train classification in Chapter 1. First, the practical railway problem is illustrated in detail in Section 1.1. This includes the common arrangement of the track installations in classification yards in Section 1.1.1, including possible variations of the track layout. Section 1.1.2 describes the way in which classification yards are basically operated, which retrospectively motivates but also results from its design. The current classification practice is then presented in Section 1.1.3 with the most important sorting methods applied today. In particular, different sorting requirements of outbound traffic are highlighted since they give rise to applying different classification methods in the first place. The main objective of a train classification process is given by the time it takes to complete it. This is illustrated in Section 1.1.4 among further secondary objectives. The section also introduces the reader to the various constraints occurring in practice. On the one hand, there are operational constraints, such as the departure times of outbound trains that must be met, and there is the infrastructure on the other. In particular, the track space is clearly limited in every classification yard, which concerns the lengths and numbers of available tracks.

The practical problem is then transfered to a theoretical model in Section 1.2. A novel representation of classification schedules, which present solutions of the train classification problem, is given in Section 1.2.3. In this representation each handling instruction of a car during a classification process is encoded by a bitstring, which uniquely determines the course of a car through the yard facilities. The encoding is applied in Section 1.2.4 to characterize feasible schedules. For a schedule to be feasible, the assigned bitstrings have to satisfy particular conditions for every pair of cars, but it suffices to impose this relation only for a certain subset of pairs of cars. The established necessary and sufficient feasibility conditions are then used to derive feasible schedules. In Section 1.2.5 this finally leads to a polynomial-time algorithm for computing feasible classification schedules of minimum length. This algorithm only considers the order requirements of the outbound trains, which presents the most fundamental version of the train classification problem.

Chapter 2 takes a closer look at the lengths of the involved tracks as one of the most important constraints. Considering this as an individual constraint was proved to result in an $\mathcal{NP}$-hard problem in [JMMN11]. In order to still derive optimal classification schedules, an integer programming model is established in Section 2.3. Based on the schedule representation by bitstring assignments, this model covers the order requirements for feasible schedules developed in Section 1.2.4 and contains constraints that handle the setting with tracks of limited lengths. Then, a polynomial-time 2-approximation is derived in Section 2.4, including two heuristically improved variants making use of there being multiple outbound trains. All the versions of this algorithm inhere a condition that allows, though not necessarily, identifying calculated approximate solution as being optimal. The four approaches, and a further very simple heuristic algorithm, are compared w.r.t. their suitability and performance in an extensive experimental study in Section 2.5. The evaluation regards different objectives, including how frequently the necessary optimality condition applies for the different approximation variants, for real-world traffic data as well as a big number of synthetically derived classification problems.

Several other restrictions, individually as well as integratedly, are studied in Chapter 3. First, as the second spacial restriction, a limited number of tracks available for a classification process is treated in Section 3.2. This restriction of the practical problem is translated in a characteristic of binary schedule encodings. In contrast to tracks of limited length, a polynomial-time algorithm is provided for this setting and proved to be optimal. Then, a setting is dealt with in Section 3.3 in which parallel sorting procedures can be applied by dividing the yard and partitioning the set of cars according to their outbound train membership. Moreover, the operational constraints of train departure times and a partially time-dependent amount of available track space are considered. These problem variants are shown to be simple to solve optimally when regarded as sole restrictions each. Section 3.4 shows how they can all be incorporated into the integer programming approach mentioned above. The resulting model is tested in Section 3.5 on traffic data of a real-world classification yard, for which a schedule is computed that actually improves on the one that was originally applied. The improved schedule is finally analyzed in a thorough computer simulation in Section 3.5, which shows its applicability in practice.

Finally, in Chapter 4 the robust counterpart of the basic train classification problem is studied. The precise model is given in Section 4.1, which uses the general notion of *recoverable robustness*, that allows responding to disruptions with certain means of recovery to turn a solution back feasible. The recovery action specific to the problem model consists in including a restricted number of additional sorting steps, and a schedule is called *recovery-robust* if, for every allowed disruption, a feasible schedule can be obtained from it by applying the recovery action. This is defined in Section 4.2, followed by proving an optimal strategy of where and how to apply the recovery, i.e. to insert additional steps into a recovery-robust schedule. Unfortunately, finding an optimal recovery-robust schedule presents an $\mathcal{NP}$-hard optimization problem as shown in Section 4.3. Nevertheless, for a simple but realistic setting excluding extreme disruptions, a polynomial-time algorithm for computing optimal recovery-robust solutions is developed in Section 4.4. This algorithm is evaluated experimentally in Section 4.4.4, which shows how the algorithm can be applied to vary between fast classification procedures and higher degrees of robustness. The chapter closes with a generalization of the robust problem setting to partially indistinguishable cars in Section 4.5, a setting which extends the solution space in general. For the special case of singleton trains, an efficient algorithm is derived for computing optimal recovery-robust train classification schedules.