1.1 Historical Review on Methods for Rearranging Railcars

Undoubtedly, one of the most essential achievements in engineering was the development of steam-powered engines at the end of the 18th century. This invention became a driving force behind the Industrial Revolution, underpinned increases in production capacity in many industries and gave birth to the railways, a fast and cost-efficient transportation system which lent additional impetus to the industrial age.

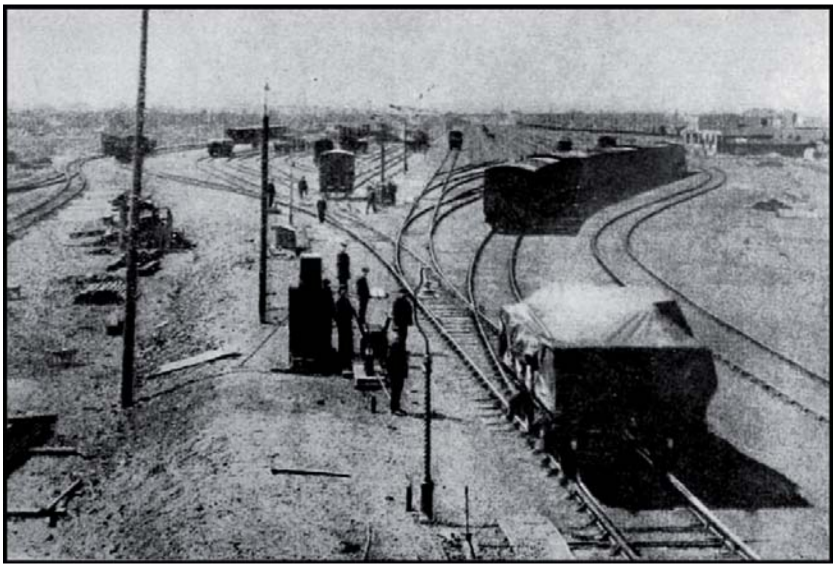

Figure 1.1. Feltham Marshalling Yard, England. *Source:* The New Zealand Railways Magazine, Volume 1, Issue 9 (February 25, 1927)

Rearranging the railcars was – and still is – one of the biggest challenges in operating railways. In addition to switching locomotives from one side of the train to the other – trains mostly shuttled between two stations in the early days – shunting was unavoidable in the case that a broken or malfunctioning railcar had to be replaced. Until the mid-19th century such rearrangements were performed at common rail stations where passengers could occasionally witness the effort workers devoted to this dangerous task. The public stations became operational bottlenecks as traffic increased; consequently, non-public *rail yards* (in other words *shunting yards*, *classification yards*, *marshalling yards*) were built, see Figure 1.1. Nowadays in these yards many inbound trains are split up, rearranged, and attached to several outbound trains at the same time.

Futhner's method is one of the oldest methods for rearranging railcars in a rail yard. According to the authors of IVIĆ ET AL. (2007) it is named after the author Harry Futhner who in 1880 was the first to apply it in practice at the Liverpool station consisting of parallel dead-ended tracks. The task was to rearrange an incoming sequence of railcars – possibly a few trains in succession – in order to form g outbound trains. For each railcar it was determined in advance with which train it had to leave, and the trains were required to depart from the station in a given order. Futhner's method is a two-step sorting procedure which requires $\lceil \sqrt{g} \rceil$ tracks. In the first phase the railcars of the 1st, $(\lceil \sqrt{g} \rceil + 1)$-th, $(2 \cdot \lceil \sqrt{g} \rceil + 1)$-th, ... departing trains are placed on track 1, the railcars of the 2nd, $(\lceil \sqrt{g} \rceil + 2)$-th, ... on track 2, and so on, see Figure 1.2. This classification allows – after pulling out all railcars in the order of increasing track numbers – a second sorting of the railcars to tracks, such that the trains can leave the station without additional rearrangements. This method presumably worked well in practice, since only a couple of trains had to be formed at the same time, i. e., $\lceil \sqrt{g} \rceil$ did usually not exceed the number of available tracks, and because the outbound trains carried only a few railcars such that the tracks were long enough for the implementation of Futhner's scheme.

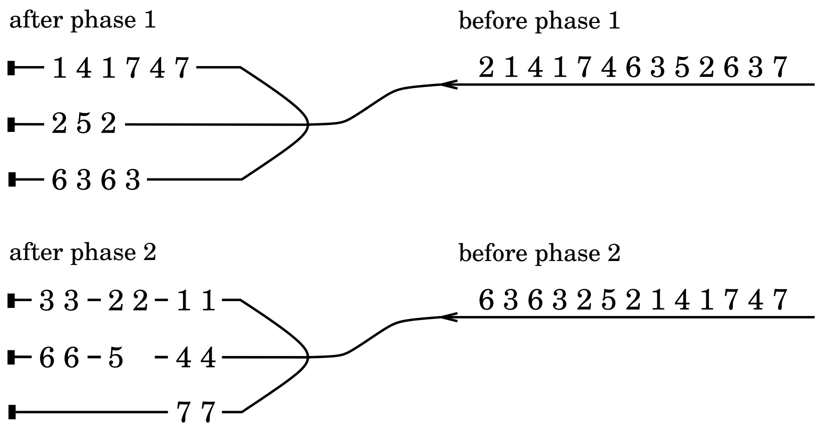

Figure 1.2. Futhner's method applied in Liverpool Station around 1880 (each number i corresponds to a railcar that has to leave with train i)

Over the course of time, similar rule-based methods for rearranging railcars were developed and applied in practice. Among the most famous are the *simultaneous*, *triangular*, or *geometric* schemes. The first articles discussing the benefits and drawbacks of various schemes – FERTIG (1927), WÖCKEL (1949), GRASSMANN (1952), FLANDORFFER (1953), BAUMANN (1959), PENTINGA (1959), KRELL (1962, 1963) – were published in the railways magazines. On the one hand, the advantage of these rule-based methods is their simplicity and transparency; out of habit the staff know exactly what to do, no matter how the arriving railcars are actually ordered. On the other hand, by not exploiting this particular order one may lose some potential for saving time and money. In other words, the sorting might be realized faster with less tracks and shunting operations with a scheme tailored to the incoming sequence of railcars. The oldest of such strategies found in the literature – see KÖNIG & SCHALTEGGER (1967), SCHALTEGGER (1967) – was introduced by the Group Operations Research at Schweizer Bundesbahnen in the late sixties of the last century. Under the direction of the mathematician Peter Schaltegger, they developed a mathematical optimization approach and implemented an algorithm in FORTRAN IV. Although they could – under certain assumptions – determine an optimal simultaneous scheme for the predicted order of incoming railcars on an Univac 1107 within seconds, there were obstacles for applying it in daily action. The problem of instantly giving the details of the computed schedule to all employees involved in the process was only one reason why optimization methods could not prevail against rule-based schemes in practice at that time. The next three decades produced little methodological progress for rearranging railcars, and from a theoretical perspective the literature – SIDDIQEE (1972), TARJAN (1972), PETERSEN (1977a,b), ASSAD (1981, 1983), DAGANZO ET AL. (1983), DAGANZO (1986, 1987b,a) – was again mainly on analyzing the effectiveness of rule-based strategies for different scenarios.

In the course of time various technical advances in rail yards created the prerequisites for convenient application of automatically generated schedules that guarantee an efficient rearrangement of specific incoming railcars. Automatic switches and brakes replaced mechanical ones. Nowadays the dispatcher most often monitors and controls the processes from the control tower, and the few people who are physically involved are connected via fast modern communication networks.

As a consequence, there is increasing interest on the practitioners side for active optimization tools that can automatically generate schedules. Before the turn of the millennium, it was rather rare for practitioners and researchers to work together in this field. However, in recent years, quite a few projects between rail operators and universities were launched. Once inspired by the application, the involved researchers were intrigued by the beauty of the underlying theoretical problems. The following selection of recent publications shows that rearranging railcars has been a hot topic from both the theoretical and practical perspectives for the last decade: WINTER (2000), DAHLHAUS ET AL. (2000b,a), LÜBBECKE & ZIMMERMANN (2000), WINTER & ZIMMERMANN (2000), CORNELSEN & DI STEFANO (2004, 2007), DI STEFANO & KOČI (2004), FRELING ET AL. (2005), LÜBBECKE & ZIMMERMANN (2005), KROON ET AL. (2006), JACOB (2007), JACOB ET AL. (2007), HANSMANN & ZIMMERMANN (2008), CESELLI ET AL. (2008), MÁRTON ET AL. (2009), EGGERMONT ET AL. (2009), BORNDÖRFER & CARDONHA (2009), HAUSER & MAUE (2010). Relevant details and results of above publications are given at appropriate places throughout the thesis. For a recent introductory survey, see GATTO ET AL. (2009).

Nevertheless, in most railway operating companies there still exists no active optimization tool as decision support for the dispatchers at the present time. One reason may be that it is hard to come up with a standard approach. The schedules that need to be generated depend highly on the particular infrastructure of the rail yard, the configuration of inbound and outbound trains, and the requested objective. Thus, methods for computing schedules of high quality have to be tailor-made to the actual situation.

In this thesis we introduce a thorough classification of many versions of such rearrangement problems. Regarding optimization methods and computational complexity, we summarize known results and present new findings for a multitude of versions. In particular, we discuss the results obtained in our research project with BASF.

1.2 Classification of Rearrangement Problems

In rail yards incoming freight or passenger trains are split up, parked, and rearranged according to destination or according to railcar construction type, see Figure 1.3. Uncertain arrival times, ad hoc changing orders of incoming railcars, the increasing number of rolling stock, sparse capacities, and financial constraints complicate the process and offer large potential for optimization.

In the above context, we provide a description of a quite general class of problems called SORTING OF ROLLING STOCK – in the following SRS for short – that cover a broad range of special applications. In general, such problems consist of three processes: arrival, parking, and departure. At the beginning an ordered *input sequence* of *units* of rolling stock (railcars, trams, complete trains, . . .) arrives at the rail yard. Then the parking process starts and the units enter the tracks of the rail yard. Here, incoming units have to be parked in such a way that at departure time the parked units can leave the rail yard in a structured *output sequence*. Note that the output sequence may contain information for several outbound trains.

The difficulty of SRS depends on the structural differences of the input sequence and the requested output sequence, as well as on the structure and flexibility of the rail yard.

Figure 1.3. One of the rail yards at the site of our practical partner: BASF, The Chemical Company, Ludwigshafen

Structure of Output Sequence As usual the incoming units are classified by a particular distinctive criterion, e. g., their destination or their construction type. As common in practice, we say that units satisfying the same criterion form a *group*.

We distinguish the following different structures of output sequences. Suppose, all positions of the output sequence are labeled, e. g., with letters, and all assigned units departing at positions with identical label are members of the same group; and, vice versa, all members from a group are assigned to positions with identical labels. In particular, the number g of different groups is the same as the number of different labels. The labels of the positions of the output sequence form certain *patterns*. For example, consider the input sequence $(2,3,2,1)$ of four units: the first and third incoming unit belong to group 2, the second to group 3, and the last incoming unit to group 1. Assume that the pattern (u,v,w,v) was requested for the output sequence. For output sequences with a **free g-pattern**, there is no fixed assignment between the g groups and the labels. The desired **free 3-pattern** (u,v,w,v) allows two configurations of the output sequence, namely $(1,2,3,2)$ and $(3,2,1,2)$. On the contrary, if there is a fixed assignment between the g groups and the labels – units departing at some position of the output sequence have to be members of a pre-defined group – we speak of output sequences with **ordered g-pattern**. If the assignment in the above example were $u \mapsto 3$, $v \mapsto 2$, and $w \mapsto 1$, then the requested configuration of the output sequence would read $(3,2,1,2)$.

We say that the output sequence has a *block* pattern, if the positions of the output sequence are labeled in a blockwise manner, that is, if the output sequence contains no subsequence of positions labeled (u,v,u) for distinct labels $u \neq v$. An output sequence with a block pattern has the structure **free g-blocks** if there is no pre-defined assignment between the g groups and the labels, and **ordered g-blocks** otherwise. Thus, if the structure **free g-blocks** is required, there are $g!$ feasible configurations (block patterns) of the output sequence according to $g!$ possible orders of the groups at departure; for the input sequence $(2,3,2,1)$ they read $(1,2,2,3)$, $(1,3,2,2)$, $(2,2,1,3)$, $(2,2,3,1)$, $(3,1,2,2)$, and $(3,2,2,1)$. In the **ordered g-blocks** case, we only get one feasible configuration of the output sequence. For the above input sequence, that is $(1,2,2,3)$ for the block pattern (u,v,v,w) with the assignment $u \mapsto 1$, $v \mapsto 2$, and $w \mapsto 3$.

In most cases, if it is required that the output sequence has one of the above-mentioned structures – **free g-pattern**, **ordered g-pattern**, **free g-blocks**, or **ordered g-blocks** – then SRS corresponds to forming one outbound train with the respective structure on one output track, see Figure 1.4. On the contrary, we say the output sequence has the structure **o-ordered g-blocks** (or **o-ordered g-pattern**) if it enables an assembly of o outbound trains on o parallel output tracks – without additional rearrangements – such that each outbound train has the desired structure **ordered blocks** (**ordered pattern**), see Chapter 3 for a more detailed description.