Advanced Methods for Railroad Station Operation Decisions: Data Analytics, Optimization, Automation

Yuan Wang
University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Urban Studies and Planning Commons

Scholar Commons Citation

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.
Advanced Methods for Railroad Station Operation Decisions:

Data Analytics, Optimization, Automation

by

Yuan Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Civil and Environmental Engineering
College of Engineering
University of South Florida

Co-Major Professor: Yu Zhang, Ph.D.
Co-Major Professor: Xiaopeng Li, Ph.D.
    Zhenyu Wang, Ph.D.
    Seckin Ozkul, Ph.D.
    Yang Liu, Ph.D.

Date of Approval:
September 23, 2022

Keywords: Anomaly Detection, Train Platforming and Routing, Symbol Recognition,
Graph Modeling

Copyright © 2022, Yuan Wang
Dedication

This dissertation is dedicated to my beloved parents, Kathy and our corgi Gubi, for their unconditional love and selfless support in my life.
Acknowledgments

I would like to express my most sincere thanks to my co-advisors Dr. Xiaopeng Li and Dr. Yu Zhang for their profound guidance and selfless help during my Ph.D. study. Their extensive knowledge and deep vision have inspired my past research and will sustainably advance my future research career. I’m also very thankful to my dissertation committee members Dr. Zhenyu Wang, Dr. Seckin Ozkul, and Dr. Yang Liu for their insightful advice during the preparation of this dissertation. From their helpful advisement, I learned to identify research opportunities, decompose problems, propose solutions, and most importantly present contributions to audiences.
Table of Contents

List of Tables ................................................................. iii
List of Figures ............................................................... iv
Abstract .......................................................................... v

Chapter 1: Introduction ......................................................... 1
  1.1 Railroad Station Anomalies Detection .......................... 3
  1.2 Train Platforming and Routing with Different Interlocking Modes .. 8
  1.3 Scientific Questions and Research Goals ...................... 19
  1.4 Dissertation Overview .............................................. 19

Chapter 2: An Innovative Huffman Forest Based Method to Detected Railroad Station Anomalies ........................................ 21
  2.1 Huffman Anomaly Detection Forest (HuffForest) ............ 22
  2.2 Sampling-Based HuffForest ........................................ 26
  2.3 Experimental Results .............................................. 28
    2.3.1 Performance on Synthetic Data ......................... 28
    2.3.2 Performance on Public Benchmarks ...................... 32

Chapter 3: Optimality Analysis of Train Platforming and Routing with Different Interlocking Modes .................................. 34
  3.1 Problem Formulation ................................................. 34
    3.1.1 TPR Decisions .............................................. 37
    3.1.2 Resource Occupations with Different Interlocking Modes .. 37
    3.1.3 Space-time Formulation .................................... 39
  3.2 Numerical Experiments .......................................... 41
    3.2.1 Solution Approaches ..................................... 41
    3.2.2 Experiment Inputs ......................................... 42
      3.2.2.1 Stations and Traffic Densities .................. 42
      3.2.2.2 Train Schedules ................................... 43
    3.2.3 Experiment Result ........................................ 44
      3.2.3.1 Sensitivity Analysis on Traffic Densities ....... 45
      3.2.3.2 Sensitivity Analysis on Station Sizes .......... 47

Chapter 4: An Automation Solution to Convert CAD Engineering Drawings into Railroad Station Models .................................. 49
  4.1 Railroad Station Symbol Recognition .......................... 49
    4.1.1 Basic Primitives and Operations ....................... 49
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.2</td>
<td>Break and Cluster</td>
<td>51</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Switch Symbol Recognition</td>
<td>53</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Track Symbol Recognition</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>Railroad Station Modeling Automation Solution</td>
<td>55</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Railroad Station Graph and Required Configurations</td>
<td>56</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Entering and Exiting Routes Search</td>
<td>57</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Summarized Procedures of Railroad Station Modeling</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>Numerical Experiments</td>
<td>60</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Experiment Goal and Data</td>
<td>60</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Validity Analysis</td>
<td>62</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Productivity Analysis</td>
<td>62</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Applicability Analysis</td>
<td>63</td>
</tr>
</tbody>
</table>

Chapter 5: Conclusions and Future Works | 66 |

References | 68 |

Appendix A: Copyright Permissions | 78 |
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Local clustered experiment result</td>
<td>29</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Local anomalies parallel to the axis experiment result</td>
<td>30</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Global anomalies parallel to the axis experiment result</td>
<td>30</td>
</tr>
<tr>
<td>Table 2.4</td>
<td>Anomalies surrounded by normal instance experiment result</td>
<td>31</td>
</tr>
<tr>
<td>Table 2.5</td>
<td>Comparison of HF, iF and LOF on 12 benchmark datasets</td>
<td>32</td>
</tr>
<tr>
<td>Table 2.6</td>
<td>Comparison of S-HF, iF and LOF on 3 large benchmark datasets</td>
<td>33</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Table of notation</td>
<td>35</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Station profiles</td>
<td>42</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Input cases</td>
<td>44</td>
</tr>
<tr>
<td>Table 3.4</td>
<td>Experiment results</td>
<td>45</td>
</tr>
<tr>
<td>Table 3.5</td>
<td>Additional experiment results</td>
<td>47</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Binary operations between primitives</td>
<td>50</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Validity analysis on four DXF files</td>
<td>62</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Productivity analysis on four DXF files</td>
<td>63</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1 Decision-making problems hierarchy .......................... 1
Figure 1.2 Railroad (passenger) network of China - 2016 version .......... 6
Figure 1.3 Routes demonstration ........................................... 9
Figure 1.4 Three types of interlocking modes .............................. 10
Figure 1.5 Screenshot of Guangzhou Nan station CAD engineering drawing ... 15
Figure 2.1 4 synthetic datasets .............................................. 29
Figure 3.1 Railroad station example ........................................ 34
Figure 3.2 Space-time occupation demonstration ............................ 41
Figure 3.3 Optimalities on different minimum headways .................... 46
Figure 3.4 Lifts on three stations .......................................... 47
Figure 4.1 Basic primitive examples ....................................... 49
Figure 4.2 Distance operations examples .................................. 51
Figure 4.3 Angle operations examples .................................... 51
Figure 4.4 Basic primitive break examples ................................. 52
Figure 4.5 Switch symbol examples ....................................... 54
Figure 4.6 Configuration examples ....................................... 58
Figure 4.7 Platform track configuration examples .......................... 61
Figure 4.8 Anylogic model of XAB ........................................ 64
Abstract

The continued and substantial growth in railroad transportation in many countries inspires railroad operators to leverage advanced methods into better railroad operation decisions. Among all facilities, significant returns on investment can always be achieved by optimizing the operations of network nodes - junctions and stations, because stations usually form the capacity bottlenecks in the system.

There are thousands of decision-making problems in relation to the station operations and can be classified into different levels to achieve different goals. From top to bottom, high-level business strategies aim to make decisions to achieve long-term strategical benefits, such as optimizing local station functionalities to succeed in global network-wise goals. Mid-level management tactics focus on formulating mid-term optimization tactics to improve management capabilities on safety, efficiency, and operating costs. Ground-level operating decision supports are to support short-term or real-time operations such as rescheduling timetables after a disruption. Foundation-level technology capabilities provide various fundamental IT tools to support the implementation of upper levels, such as data collection, data preprocessing, and process automation.

Following this hierarchy, three problems draw the interest of this dissertation:

Firstly, to fill the gap of using scientific methods to understand station operations and identify target objects at the strategic level. A meaningful railroad station anomalies detection problem attracted the attention of this dissertation.

Detecting railroad station anomalies is a critical task prior to segmentation and making optimization decisions for each cluster. Three types of anomalies caused by the specialty of railroad operations bring the existing methods non-trivial challenges in detection accuracy and efficiency. To tackle these tasks, this dissertation proposes a novel anomaly detection
method named Huffman Anomaly Detection Forest (HuffForest) to detect station anomalies in performance, which leverages Huffman encoding to measure abnormalities in certain railroad scenarios with high accuracy.

This method establishes a Huffman forest by constructing trees from the perspective of data points and subsequently computes anomaly scores of instances considering both local and global information. A sampling-based version is also developed to improve scalability for large datasets. Taking advantage of the encoding mechanism, the proposed method can effectively recognize the underlying patterns of railroad stations and detect outliers in various complicated scenarios where the conventional methods are not reliable.

Experiment results on both synthesized and public benchmarks are demonstrated to show the advances of the proposed method compared to the state-of-the-art isolation forest (iForest) and local outlier factor (LOF) methods on detection accuracy with an acceptable computational complexity.

Secondly, after the target stations are properly selected, this dissertation is motivated to drill down to the tactical level of decisions to maximize station capacity and scheduling reliability, where a train platforming and routing (TPR) problem with different interlocking modes is identified.

The TPR problem is to decide train operations within stations after the network-wise train schedules are determined. A feasible TPR plan requires both platform and route conflict-free, where the avoidance of route conflict is controlled according to three interlocking modes. Although the TPR problem is widely studied, none of them did a serious investigation on the optimality impacts of different interlocking modes in the TPR. Therefore, this dissertation introduced and formulated a space-time version of TPR considering three interlocking modes and subsequently conducted numerical experiments to analyze the optimality differences under each mode. Based on the experimental findings, engineering practical suggestions are also provided. In summary, the experiment results showed that both route-locking sectional-release and sectional-locking sectional-release modes sig-
nificantly outperform the route-locking route-release mode. And among them, using the route-locking sectional-release mode can bring notable benefits to large stations with high-density volumes while using the sectional-locking sectional-release mode can always provide outstanding outcomes over various station and traffic settings.

In the meanwhile of addressing the above issues, a fundamental technical problem at the tech-foundation level arises - the manual modeling approach often costs engineers significant efforts and notably limits the generality and extensivity of many advanced methods.

However, creating a high-fidelity railroad station model to match the physical details of hundreds of tracks and switches is never a trivial task. With the widespread of CAD technology, many stations are drawn proportionally into two-dimensional DXF files. Thus, this dissertation proposed a framework to efficiently convert DXF files into meaningful station models. The proposed framework consists of two phases (1) converting graphic basic primitives without explicit engineering interpretations into recognizable railroad symbols and (2) modeling undirected railroad station graphs with necessary configurations such as endpoints and routes. Subsequently, the proposed framework is developed into a GUI application with minimal user interaction, and the validity, productivity, and applicability are tested at several real-world passenger stations in Asia.
Chapter 1: Introduction

The continued and substantial growth in railroad transportation in many countries indicates that railroad operations optimization remains a vital issue for decision-makers [67]. While seeking to maintain acceptable levels of service, railroad operators must make the best possible use of available infrastructures and capacities. Significant returns on investment are always achieved by optimizing the operations of network nodes - junctions and stations - that usually form the capacity bottlenecks in the system[5].

Operating a station efficiently as well as economically is a complicated and comprehensive challenge. There are thousands of station-related decision-making problems that can directly impact operating performance. For example, which station should be assigned with a higher managerial priority? Should advanced interlocking equipment be invested? What is the return on investments (ROI)? How to reduce human work by using automation techniques?

Figure 1.1: Decision-making problems hierarchy
Because of the potential benefits, many studies developed various applications to leverage advanced methods into better railroad station management decisions. In most cases, station-related decision-making problems are classified into different levels with different goals. From top to bottom, as can be seen in Figure 1.1, there are usually four levels.

- **High-level business strategies** aim to make decisions to achieve long-term strategical benefits, such as optimizing local station functionalities to meet global network-wise goals.

- **Mid-level management tactics** focus on formulating mid-term optimization tactics to improve management capabilities on safety, efficiency, and operating costs.

- **Ground-level operating decision support** is to support short-term or real-time operations such as rescheduling timetables after a disruption.

- **Foundation-level technology capabilities** provide various fundamental IT tools to support the implementation of upper levels, such as data collection, data preprocessing, and process automation.

Following this hierarchy from top to bottom, three problems draw the attention of this dissertation:

1. A meaningful railroad station anomalies detection problem attracted the interest of this dissertation. By studying this problem, this dissertation aims to fill the gap of using scientific methods to understand station operations and identify target objects at the **strategical level**.

2. After the target stations are properly selected, this dissertation is motivated to drill down to the **tactical level** decisions of stations and notices a train platforming and routing problem with different interlocking modes. In this problem, this dissertation intends to seriously investigate the pros and cons of different interlocking modes when they are involved in maximizing station capacity and scheduling reliability.
3. In the meanwhile of addressing the above issues, a fundamental technical problem at the Tech-foundation level arises - the manual modeling approach significantly limits the generality and extensivity of many advanced methods, which encourages this dissertation to develop an automation framework to create station models validly and productively.

1.1 Railroad Station Anomalies Detection

Subject to resource constraints, such as expertise, investments, and management capabilities, it is usually regarded as engineering impractical to make each station a dedicated optimization strategy. A more realistic and cost-effective way is to segment stations into several classes according to their performances and make diversified strategies for each class. For instance, the Chinese railroad has thousands of stations with different features such as trains throughout, origination train count, destinations train count, passenger population within 50 miles, passenger volume, platform count, terminal house squares, connecting direction count, LBS information, etc. These stations are usually divided into three levels (e.g., premier-class, first-class, and second-class) according to traffic or four levels (e.g., super large, large, medium, and small) according to the number of platforms [20, 58]. Then, decision-makers will make each class different management requirements, objectives, and optimization strategies.

Obviously, the quality of segmentation will directly contribute to optimization performance. Regardless of segmentation techniques, one critical but challenging part is to detect outliers prior to segmentation. These outliers are usually special cases that can hardly be covered by any general strategy and should be analyzed case-by-case and treated delicately. For example, the Wuchang rail station is located at the intersection of four high-volume rail lines and has an extremely high traffic volume but a very limited number of platforms. On one hand, if it was clustered into the high-volume class, the optimization strategy may not be effective due to the limited infrastructure. On the other hand, if it was included in the
few-platforms class, the low priority setting to this class will bring many significant operation troubles due to the high-volume fact. Furthermore, this station is an isolated case caused by its location and historical reality making it managerial unnecessary to create a particular high-volume-but-few-platform class. And hence, it should be identified as an anomaly to receive a dedicated treatment.

Thanks to pioneer studies, many great measurement tools have been developed to provide insights into railroad performance for decision-makers [69, 29, 56]. These measurement frameworks provide various attributes of station performance. Also, many studies proposed various classification methods from different perspectives, ranging from traffic volume, and passenger frequency to facility conditions [71, 21, 75, 41, 42]. However, to the best knowledge, no studies focus on detecting station anomalies considering both traffic and infrastructure features.

Anomaly detection is a long-existing research topic, which focuses on identifying the data points (usually referred to as instances) whose distributions differ from the majority. Since anomaly usually implies important information, this technique is widely utilized in a variety of fields to discover outliers, thereby recognizing the abnormal phenomenon, which significantly helps the researchers in multiple real-world applications such as avoiding potential risks and capturing new observations [32].

Accurately and efficiently detecting the anomaly is usually a non-trivial task and a large volume of study has been devoted to this line of research. The existing approaches for anomaly detection can be classified into supervised approaches and unsupervised approaches, where unsupervised approaches can be further divided into distance-based methods, density-based methods, and model-based methods. Since large railroad operators usually manage thousands of rail junctions or stations [4], which makes the labels of anomalies difficult to be obtained, this dissertation focuses on unsupervised approaches in this dissertation.

Most of the existing distance-based anomaly detection techniques identify the abnormal instances based on the assumption that an outlier lies far away from the major data dis-
tribution, and is consequently subject to a greater distance than most of the others [59]. However, this series of techniques are proven to be effective only for certain types of outliers since they only focus on the global view of the dataset [10, 57]. Therefore, their detection capability is limited for more complex structures that widely exist in the real world.

In order to overcome these drawbacks of distance-based techniques, density-based measurements have been proposed to improve the capability of recognizing the “local” outliers that lie in the low-density area [10]. Although this type of methods is proven to be more effective on local outliers, there are still unsolved issues. For instance, low density does not always imply outliers especially when it is measured considering only the local context [33]. Furthermore, both density-based and distance-based methods may easily fail if there exist regions with different densities in a dataset [33].

Apart from distance-based and density-based methods, the machine learning algorithms originally developed for other tasks have also been leveraged for anomaly detection as model-based detection methods, such as support vector machine (SVM) and random forest [51, 54]. However, their abilities in anomaly detection are in fact a byproduct of the mechanisms designed to fit other scenarios and have not been optimized to detect anomalies [33]. This induces the limitation that the efficacy of a model-based method on anomaly detection critically depends on the similarity between the current and original tasks.

In contrast to the aforementioned model-based techniques, the isolation tree (iForest) is originally designed for anomaly detection. It constructs isolation trees that split the instances based on attribute values [31, 33]. Compared to normal instances, the outliers are more likely to be separated in the branches close to the roots. The isolation trees are subsequently assembled to form an iForest, which measures the level of abnormalities of each instance based on the average length of paths. Compared to the existing distance-based and density-based methods such as ORCA [8] and LOF [10] as well as other model-based methods such as one-class SVM [51] and random forest [54], iForest has demonstrated its advantages in both detection accuracy and efficiency.
Although iForest is recognized as a state-of-the-art method, there still exist several types of anomalies caused by the railroad station operations that cannot be effectively detected by it. Since stations are nodes in the railroad network that are connected by lines, some of their attributes are not independent of each other and highly impacted by the network structure. Figure 1.2 presents the simplified passenger railroad network of China, where colored lines represent the rail lines and dots represent the stations. Three commonly seen scenarios are as follows:

1. **Local clustered anomalies**, which are the local anomalies that form a cluster. Square marks in Figure 1.2 represent five stations on Hainan island, where they compose a
local network aiming at satisfying local demands and have a very weak connection to the mainland network due to the channel. They are clustered locally because their distributions of traffic and demand attributes are similar to each other and identified as anomalies since these distributions significantly differ from the majority.

2. **Axis paralleled anomalies**, which are the anomalies with identical values in some attributes, thus parallel to the axis. Triangle marks in Figure 1.2 represent four stations in the Tibet part of the Qinghai-Tibet line, which is located at the edge of the rail network. Four stations are parallel to the traffic axis since they are in the same line without any other branches and share the exact same traffic distributions, regardless of their possible variation in demands. They are further considered anomalies since their traffic is much lower than the majority.

3. **Anomalies surrounded by normal instances**. Since most of the China high-speed rail lines were newly built rather than upgraded from old lines, the function of stations is very distinguishable that focusing on either high or regular speed trains, and the distribution of average speed passing through stations follows a scattered pattern - either low or high. However, four stations with the diamond mark in Figure 1.2 are conjunctions of both high and regular speed lines and their functionalities are hybrid for both types of trains so that their average speed distributions concentrate in the middle between low and high. So, they can be regarded as anomalies (with concentrated features) surrounded by normal instances (with scattered features).

Due to the importance of these anomalies in real-world scenarios, efforts have been devoted to discovering the corresponding solutions. The work [32] has involved a split-selection mechanism in the original version of iForest and transformed it into SCiForest for the detection of local clustered anomalies. The work [6] leverages the advantages of both iForest and distance-based methods by adopting nearest neighbor distance in the space partitioning framework, which can successfully detect both local clustered anomalies and the anomalies
surrounded by normal instances with high efficiency. Despite the advances of recent research, no existing method can tackle the aforementioned three issues simultaneously.

1.2 Train Platforming and Routing with Different Interlocking Modes

Train scheduling (TS) planning or timetabling is one of the most important planning works in railroad operations that will directly determine the quality of resource utilities and operating efficiencies. Most passenger TS follows a cyclic rule, that the same pattern in a timetable is repeated every cycle (one day) in a planning horizon (couple months). And a T’s planning usually includes two levels: the company-level train scheduling department will first come up with a feasible network-wise train schedule determining the arrival and departure schedule of trains at stations. Subsequently, the station management team will be responsible for creating detailed station-level operation plans given the network-wise TS. Because stations - as nodes in the network- are the most valuable assets in the network that usually form the capacity bottlenecks in the system [5]. Significant returns will often be achieved through investments in optimizing the station-level schedules which can directly affect the safety and efficiency performance.

This crucial task of optimizing station-level operation plans rises and is often referred to as Train Platforming and Routing \((TPR)\) problem with scheduling adjustment allowed. In most passenger stations, a TPR consists of three sub-problems:

- \textit{Train Platforming} \((TP)\) problem to assign trains to platforms following various safety and business rules
- \textit{Train Routing} \((TR)\) problem to determine the train routes inside the station
- \textit{Train Schedule Adjusting} \((TSA)\) problem to make a minor deviation (e.g., 5 minutes) of arrival and departure time from the original TS so that all trains can be platformed and routed validly

TPR planning is never a trivial task and, in contrast, one of the most time-consuming works in practice. Study [14] pointed out that it can take 15 days for an expert planner
to compose a medium-scale TPR plan, which usually accounts for 30% of overall train scheduling works since planners should handle several hundred, or over a thousand, trains per cyclic period and various complex safety and business constraints should be followed study [15].

Mandatory safety rules such as no conflicts being allowed on switches and platforms should be repeatedly checked during the whole planning process, where the most challenging part is to avoid route conflicts in inbounding or outbounding areas.

Figure 1.3: Routes demonstration

The basic route conflict is defined as two trains occupying the same switch during an overlap time interval. But in practice, route conflict is defined as the time interval not long enough between one train using one route and other trains using its conflicted routes, where conflicted routes are identified as they have shared switches. Figure 1.3 presents an illustration of two conflict routes, where switches are marked with three digits numbers. In this figure, Route 1 passes switches 115,121,131,153 while Route 2 passes 157,147,133,121,119, and they are conflicted at Switch 121. The safety rule requires a minimal interval time $\lambda$ between the use of the two routes, meaning if Route 1 is used by a train at time $t$, Route 2 wouldn’t available until time $t + \lambda$. Due to limitations of manual planning capability, assuming the train length, train speed, and route length are constants, planners usually default the minimal time interval between using one route and all its conflicted routes are the same,
Figure 1.4: Three types of interlocking modes

(a) Route-locking route-release

(b) Route-locking sectional-release

(c) Sectional-locking sectional-release
and then set only one parameter $\lambda$ for each route to assure the safety to all its conflicted routes.

However, in actual operations, the avoidance of route conflicts is controlled by the modern electric interlocking systems, including three interlocking modes to control the lock and release of switches: route-locking route-release (RLRR), route-locking sectional-release (RLSR) and sectional-locking sectional-release (SLSR). Continuing on the example in Figure 1.3, Figure 1.4 demonstrates the locking and release patterns of switches under different modes, where the x-axis is the time horizon, the y-axis represents each switch and the length of each bar shows the time interval between locking and release of switches. The RLRR mode requires all switches in a route are locked at the beginning one train enters this route till the point train completely passes the last switch in this route. Figure 1.4a shows the switches locking time interval under RLRR mode, where all switches are locked immediately when Train A enters Route 1 at time $t_1$ and released all together when Train A passes the last switch at time $t_3$. In this mode, the earliest time Train B can start to enter Route 2 is $t_2$ and the minimal time interval $\lambda = t_2 - t_1$. In fact, the $\lambda$ is only determined by the duration Train A occupies Route 1, and the rules used in manual planning are actually based on this rule. The total number of $\lambda$ parameters needed to check route conflicts equals the total number of routes.

In RLSR mode, all switches are locked at the beginning one train enters this route and will be released sequentially at the time train completely passes each switch. In Figure 1.4b, all switches are locked at $t_1$ when Train A enters Route 1 and sequentially release switches 115, 121, 131, and 153 when Train A completely passes each one. After switch 121 is released, Route 2 is ready for Train B to use. So the interval time $\lambda = t_2 - t'_1$ is smaller than that in the RLRR mode. Be noted that since the minimal interval time is determined by both travel duration on each route and positions intersected with its conflicted routes, the total number of $\lambda$ parameters needed to check conflicts equals the squared number of routes. In fact, this mode is commonly implemented for most passenger stations [22].
The SLSR requires each switch in a route to be locked at the time one train actually enters it and released at the time the train completely passes it. Figure 1.4c demonstrates the locking and release pattern for both Train A and Train B, where both trains can enter their route even simultaneously as long as they don’t have an overlap of duration on occupying switch 121. In other words, the model allows one train to pass the intersection points in front of other trains that are still using its conflicted routes. In this case, the \( \lambda \) can even equal zero and the logic of checking minimal interval time is not effective anymore. Although this mode sounds a little ideal and risky in real operations, with the development of Automatic Train Control (ATC) and other precision interlocking technologies, it is still theoretically valid and may be widely implemented in the future.

Comparing the three modes, this dissertation can obviously see that the RLRR mode is simpler to be considered in planning work and has higher compatibility than others that if one plan is determined under this mode, it can also be used for other twos, whereas RLSR mode is only compatible for itself and SLSR. However, on the other hand, the suboptimality of defaulting to the first mode is also very intuitive: RLRR mode sets a longer duration of occupying switches than trains actually use and leads to potential resource utilization redundancies.

In fact, from the perspective of engineering convenience, the choice of RLRR mode is a compromise between safety concerns and resource utilization since the manual planning approach can hardly satisfy complex precision control requirements that considering RLSR or SLSR mode may bring hidden safety violations.

However, it doesn’t mean other choices are impossible in practice. With the development of advanced planning technologies such as automated planning tools or digital-twin-based conflict check techniques, planning capability can be hopefully enhanced with more precision control functionalities to cover both safety and utilization needs.

A more reasonable decision of mode choice should be return-oriented. As long as the benefits of RLSR and SLSR are significantly attractive, the investment in technology and
management should be sustainably supported. Thus, this dissertation is motivated to provide complete evidence of optimality gaps of three modes and assist in understanding the benefits of leveraging advanced techniques and modifying the safety rules in planning work. The traffic capacity and operations at conjunctions were a long-existing topic in the area of road transportation [46, 45]. And being specific in railroad transportation, references [35, 13] provided a broad overview to present commonalities and diversities of different models and methods related to TPR, and categorized railroad optimization modeling problems into three levels of modeling: *strategical, tactical* and *operational*.

Strategical level modeling aims at optimizing TPR for a range of potential timetables to assist decision-makers in understanding station capacity and making infrastructure change decisions. This level is usually concerned with main constraints and has lower requirements on feasibilities and efficiency, more information can be seen from reference [27, 26, 44, 38, 39]; tactical level modeling is to generate platforming, routing, and scheduling plans for a middle-term period that are used as fundamental plans to guide practical operations. This level includes more side constraints that will significantly affect later operations and has higher requirements on optimalities and feasibilities but medium requirements on efficiency; operational level modeling focuses on deciding the adjustment of plans according to real-time operations when tactical plans cannot be fulfilled [16, 74, 48]. This level has the highest requirements on efficiency although may sacrifice some optimalities. The problem discussed in this dissertation is classified into the tactical level.

There is a batch of early studies that considered different versions of tactical TPR. Reference [14] simplified the platform assignment problem as \(k L\)-list \(\lambda\) coloring of graphs assuming the routing and schedule time cannot be changed and [9] extended this work to an integer programming version. [78, 77, 76] proposed routing version of TPR, where trains with their routes were defined as nodes and these being pairwise conflict-free is packed into a “safety set”. [15] used a heuristic analogous of mimicking the manual planning process allowing time deviations but assuming the route to platforms is uniquely determined.
Combining all aspects together, [12] proposed a general version of the TPR problem, considering platform assignment, routing, and time deviation integrally. This study created a pattern-incompatible graph, where nodes correspond to patterns formed by enumerating all possible combinations of arrival time, paths, assigned platforms, and departure time of each train, whereas the edges connect pairwise incompatible, a.k.a conflicted, nodes, such as occupying the platform or conflicted route during an overlap time interval. This study also considered a one-time cost as long as a platform is used in the objective function and introduced a dynamic threshold to control the route conflicts. In order always to return a solution, it introduced dummy platforms where the objective function tends to find the optimal trade-off between dummy assignment and time deviation.

Although possible extensions were addressed by assuming the time dimension to be continuous [53] or optimizing the platform choice integrated with the timetable [63], [12] can still be regarded as the closest one to this dissertation.

To the best knowledge, none of the studies above seriously addressed the optimality analysis for different interlocking modes at the planning stage. In addition, on top of the formulation proposed in [12], a modified version is also needed to describe conflicts with various interlocking modes and avoid generating the exponentially growing pattern-incompatible graph.

Railroad stations, as nodes in the railroad transportation network, play a significantly critical role in forming the capacity bottlenecks in the system and attract remarkable attention from both academia and industry. Understanding the characteristics of stations (e.g., evaluating a station’s capacity and waiting time) and optimizing their operations are critical to maximizing the system performance while minimizing the system-wide cost. Accordingly, many simulation and optimization methods have been proposed to investigate capacities and optimize operations for various types of railroad stations [62, 74, 7, 36].

Prior to implementing these products, a common fundamental challenge is creating digital twins of physical stations, a.k.a railroad station modeling. For instance, to build a high-
fidelity simulation model for a station, a digital rail network model should be established to match the physical details of the station.

Modeling a railroad station is never a trivial task since complex railroad stations usually consist of hundreds of tracks and switches and dozens of railroad signals. And it usually takes several weeks of effort for an expert to manually create a station model to match the physical details of the station, which notably limits the generality and extensivity of many methods.

Figure 1.5: Screenshot of Guangzhounan station CAD engineering drawing

Digital engineering drawings are outputs of the Computer-Aided Design (CAD) technology, which is used to convey high-precision geometrical information about an object with standardized entities. Further, a powerful Drawing Exchange Format (DXF), developed by AutoCad, unified data formats for reading and writing and thus ensured data interoperability among various CAD platforms.

Thanks to the widespread of CAD technology, most stations were drawn proportionally into two-dimensional DXF files, which inspires this dissertation to utilize DXF files to automate the modeling processes of stations. Figure 1.5 shows a screenshot of the Guangzhounan station drawn with CAD tools.
However, a DXF drawing is formed by a set of graphical vectorial entities including primitives such as points, lines, arcs or basic meaningful elements such as circles, and rectangles, without explicit engineering interpretations. For example, a line can be represented by the coordinates of two endpoints, and a circle is represented by the coordinate of one point (center) and one length (radius).

Therefore, advanced methods have to be developed to translate graphic entities into recognizable symbols with obvious physical meanings to engineers. To be more specific, the task consists of two steps (1) *symbol recognition* to recognize vectorial symbols to generate station objects and (2) *modeling automation* to construct engineering relations between recognized objects.

Recognizing vectorial symbols efficiently and accurately is a long-existing challenging task because of the vast variety of symbols available in many domains. Reference [28] categorized methods into two major categories by the way of utilizing information from the graphs: statistical methods and structural methods.

Statistical methods utilize statistical or machine learning techniques to generate features according to values and distributions of pixels of images and then calculate the similarities between patterns and target symbols [43, 65] or train learning-based models to predict the probabilities of matching target symbols [30, 19, 23, 37, 47]. Since the accuracy of statistical methods is significantly impacted by the way of creating features, many references studied various feature engineering methods [68, 64, 73, 18, 55].

Although these methods have been proven effective for images, their common drawbacks are very obvious training a high-accuracy statistical model requires a large number of samples. However, railroad stations can only provide a limited number of CAD samples but require a very high recognition accuracy, sometimes even requiring absolute correctness. Besides, statistical methods tend to filter out small detail changes (pixel level) that are unfortunately inconvenient in our context [50]. Thus, these methods are not well adapted to the problem of station CAD symbol recognition.
Instead of considering the statistics of symbol graph features, structural methods include two parts. The first step is to describe source symbols by defining basic entities and their relations. For example, a railroad switch symbol is described as three lines intersecting at one point, and a railroad track is described as a set of consecutively connected lines where the connection point is not a switch. Then the second step is to identify target symbols matching descriptions.

The description phase also consists of two tasks: defining entities and using entities and their relations to describe symbols. First, although a few studies proposed different methods to define entities such as references [1, 72], this dissertation uses the entity definition in DXF files for engineering convenience and focuses on defining the entity relations in a clear and efficient way. Secondly, in the relation description part, most studies utilized the topological graph to describe symbols where entities form the nodes and relations between pairwise entities form edges such as constraint network [3, 2], attributed relational graph (ARG) [50, 11, 17], region adjacency graphs (RAG) [34], spatial relations [50]. There are also some studies that were not based on graphs such as bag-of-relations [49] and deformable templates [60].

These techniques tend to provide comprehensive solutions to recognize symbols. However, none of them are designed specifically for railroad scenarios and have notable engineering overheads to be implemented. Thus, taking advantage of existing techniques, engineering practical and convenient methods that can be better applied to railroad stations should be studied and developed accordingly.

After the symbol recognition problem is properly addressed, there are still many non-trivial works to ensemble independent symbols to a railroad station model - build connections between symbols according to their engineering interpretation.

There are various types of models and applications related to railroad station operations. For example, Petri net was often utilized to simulate station operations. Study [66] used Petri net to validate the signaling and interlocking design of railroad stations and reference [25]
used a hybrid Petri nets-based simulation model to evaluate the design of railroad stations. In addition, graph models were often used to optimize station operations as well. For instance, studies [14, 70, 76] modeled stations into graphs where unique routes were nodes and pairwise routes without conflicts were connected as arcs to satisfy route conflict-free constraints. Caprara [13] used a pattern-incompatible graph to solve the platforming problems where nodes were train decision patterns (train schedules, route, and platform selections) and arcs indicated pairwise patterns being incompatible.

These models demonstrated superior as well as inferior capabilities in different scenarios. Petri net is excellent at modeling discrete even simulation but inconvenient to be used in optimization models. And route-based graph models are practical in solving integer programming but not straightforward nor detailed enough to represent stations if they are applied in simulation. Later Wang [62] used an undirected graph to present railroad stations with rich physical details such as track length. In this model, switches formed the nodes and tracks were undirected arcs connecting switches. In order to detect route conflicts, all entering or exiting routes were additionally defined using a sequenced set of nodes so that two conflicted routed can be identified if they have shared nodes.

The above studies provided broad views of railroad station models. However, none of them gave details on efficient means of converting real-world stations to mathematical models. Especially for the undirected graph presented in reference [62], although this modeling way is straightforward to be understood and can contain enough physical details of stations, it will cost significant efforts to build from scratch manually that limiting its scalability and extensity. But on the other hand, this dissertation can also see its vast potential if efficient support of modeling automation techniques is developed properly. Therefore, this dissertation focuses on converting DXF files into undirected graph models of railroad stations.

In terms of the automation application, reference [52] presented an automated platforming and routing tool which can solve optimization problems for all 530 stations in Belgium. In this study, station details such as lines, platform, and routes were infrastructure inputs
and should be configured by users in advance. Although it didn’t point out the cost of configuring infrastructure inputs, significant efforts were likely needed.

Overall, to the best knowledge, no studies provided solutions for efficiently converting DXF files to railroad station models.

1.3 Scientific Questions and Research Goals

Based on the research gaps revealed in the above sections, this dissertation aims to address the following questions or goals:

1. How advanced data analytics methods should be developed to detect railroad station anomalies which can tackle the limitations of the existing anomaly detection techniques when they are utilized in similar scenarios of rail station anomaly detection? What is the time efficiency of the developed method and how can it be improved to provide a scalable solution for large datasets?

2. What are the impacts of different interlocking modes on the optimality of train platforming problems? What are their sensitivities to various station sizes and volume levels? How the state-of-the-art approach can be improved so that the route conflicted can be easier formulated?

3. How advanced techniques can be developed that can better recognize railroad symbols? How automation frameworks can be developed to replace/supplement the manual modeling process for stations?

1.4 Dissertation Overview

The remainder of this dissertation can be summarized as follows.

In Chapter 2, (1) this dissertation proposes an innovative Huffman forest anomaly detection method (HuffForest) which can tackle the limitations of the existing anomaly detection techniques when they are utilized in similar scenarios of rail station anomaly detection. (2)
A sampling-based version of the HuffForest method is developed to improve time efficiency and provide a scalable solution for large datasets. (3) Experiments are conducted to validate the proposed Huffman forest-based anomaly detection methods, where the proposed methods outperform existing methods on detection accuracy in 14 public benchmarks with acceptable computational cost.

In Chapter 3, (1) this dissertation investigates instructive findings on the optimality gap of three interlocking modes, that are important to engineering practice in making decisions on modifying the planning rules and leveraging advanced technology capabilities. (2) A simpler space-time formulation is proposed by modifying the state-of-the-art techniques, that can describe router conflicts more straightforwardly and avoid generating the exponentially growing pattern-incompatible graph (3) Thoroughly numerical experiments are conducted to provide evidence.

In Chapter 4, (1) this dissertation proposes a symbol recognition and modeling framework that is better adapted to railroad practice and can efficiently convert DXF files into undirected graph models of railroad stations with minimal human interactions. (2) experiments are conducted to validate the proposed method and a commercialized system is developed and tested at several large passenger stations in Asia.

Finally, in Chapter 5, this dissertation is closed, the major findings are summarized and interesting future works are discussed.
Chapter 2: An Innovative Huffman Forest Based Method to Detected Railroad Station Anomalies

In order to overcome these difficulties, this dissertation has reconsidered the characteristics of outliers and find distance still an effective measurement of abnormality level if utilized in an appropriate way. Thus, the proposed method first computes the distances between each pair of instances and subsequently evaluates the abnormality of each instance using both global and local information extracted from the distances. For an instance, the global information refers to its abnormality level observed by other instances based on the distances and the local information refers to the abnormality levels of other instances from the perspective of itself, which can reflect the abnormality level of its own in return. In order to effectively transfer the distance information into abnormality levels, Huffman encoding is employed to accurately identify the remote distances, which usually lie in the rare region of all distances. Thus, a Huffman tree is built for each data instance and thereby forming a Huffman forest for the dataset. Subsequently, an anomaly score is computed based on the encoding depths of the Huffman trees. Considering the nature of the proposed method, it is named Huffman Anomaly Detection Forest (HuffForest).

It should be noted that detection accuracy is usually given a much higher priority than computation complexity since rail stations are critical assets for railroad companies and optimization investments tend to be massive and long-term. Therefore, although the proposed method is limited by the high time complexity since the distances between all pairs of instances need to be computed, the strengths of detecting anomalies with higher accuracy can still be taken advantage of. And in order to minimize the weakness of time efficiency, a sampling-based version of HuffForest is also developed. It computes the anomaly score of an

---

1 This chapter is derived in part from an article published in Sensors [61]
instance using a subset of instances created by an innovative stratified sampling mechanism. This significantly improves the scalability for processing large datasets without sacrificing detection accuracy.

In order to validate the efficacy of HuffForest, four synthetic benchmarks are created to represent the above facts of railroad networks. These benchmarks contain local clustered anomalies, axis paralleled anomalies, and anomalies surrounded by normal instances, respectively. While these scenarios cannot be simultaneously handled by existing anomaly detection techniques, HuffForest can successfully detect the anomalies for all these benchmarks. Furthermore, in order to better demonstrate the strength of the proposed methods, the original and sampling-based HuffForest are also compared with LOF and iForest on 15 public benchmarks, which shows a superior detection accuracy in 14 of them with acceptable computational costs.

2.1 Huffman Anomaly Detection Forest (HuffForest)

Consider a set of instances denoted by \( \{x_i\}_{1 \leq i \leq n} \) while \( x_i \in \mathbb{R}^m \), where \( m \) is the number of dimensions of each instance. The target of this work is to identify abnormal instances based on topological information. Hence, the distances between all pairs of instances are computed while the distance between \( x_i \) and \( x_j \) is denoted by \( d_{ij} \). Other than directly computing the distances, this dissertation first normalizes the instances to mitigate the impact induced by the differences in distributions between each dimension. The normalized version of an instance \( x_i \) is represented by \( \tilde{x}_i \), in which the \( k \)-th dimension is computed by

\[
\tilde{x}_{ik} = \frac{x_{ik} - \mu_k}{\sigma_k}, \forall 1 \leq i \leq n, 1 \leq k \leq m
\]  

(2.1)

where \( \mu_k \) and \( \sigma_k \) are the mean and standard deviation of dimension \( k \) over the dataset, respectively. Subsequently, the distance between instances \( i \) and \( j \) is computed as the Euclidean
distance between $\bar{x}_i$ and $\bar{x}_j$,

$$d_{ij} = \|\bar{x}_i - \bar{x}_j\|_2, \forall 1 \leq i, j \leq n$$  \hspace{1cm} (2.2)

The most critical part of the proposed method is the way to effectively utilize the distances. Based on the assumption that the abnormal instances are those remote from the major distribution, a natural idea is to measure the “remote level” of each instance and identify the abnormal ones accordingly. Unlike the existing distance-based approaches, this work identifies the abnormal instances using a voting-based method since this dissertation has the following two observations.

- For a normal instance, the range of distances from it to others is relatively wide since normal instances can be close to it, whereas abnormal instances can be far away from it.

- For an abnormal instance, the range of distances from it to others tends to be narrow since most instances are normal ones that are usually remote from it.

Based on the aforementioned observations, each instance can create its own view of the others in the dataset and the abnormality level of an instance can be measured using both global information and local information. For an instance, global information refers to the evaluation of it created by the others and local information refers to its evaluation of the others in the dataset. Global and local information can be jointly considered to form the abnormality measurement for each instance.

At this step, a crucial task is to find an appropriate representation of the measurements. In order to reach this target, Huffman encoding [24] is employed in this work since it can use the encoding length to represent the frequency of appearance. Given a set of symbols, Huffman encoding represents them using variable-length codes based on frequency-of-occurrence for efficient information compression such that a symbol with higher frequency is represented by a code with a shorter length [24]. In Huffman encoding, each symbol to be encoded is
treated as a node and the two nodes with the minimum frequencies are iteratively grouped as a new node, which forms a binary tree structure, referred to as “Huffman Tree”. Following this logic, a longer distance can be projected to a longer encoding length since it is usually a rare event. Thus, in order to measure the abnormality level of an instance, a natural idea is to form a Huffman tree from the perspective of a specific instance (termed as reference instance), in which all the instances in the dataset except the reference instance are treated as symbols. For a dataset, Huffman trees are generated from the perspective of each instance, which forms a Huffman forest that combines the evaluation of the dataset illustrated by each tree.

Considering the original definition of the Huffman tree, this dissertation uses frequencies-of-occurrence derived from distances as the weights of instances. To be specific, while constructing a Huffman tree based on a reference instance, the instances whose distances to the reference instance lie in a high-density range are those with high frequencies-of-occurrences. In order to implement this idea, the distances need to be discretized to appropriately compute frequencies-of-occurrence before being encoded using Huffman forest. In order to guarantee that the distances can be discretized using a unified criterion, this dissertation further normalizes the distances for each potential Huffman tree as in Eqn. (2.3) such that it is distributed between 0 and 1.

$$d^*_i(x_j) = \frac{d(x_i, x_j)}{\sum_{j=1}^{n} d(x_i, x_j)}, \forall 1 \leq i, j \leq n$$  \hspace{1cm} (2.3)

where $i$ refers to a reference instance. Subsequently, $d^*_i(x_j)$ are discretized using a stride of 0.001. Subsequently, this dissertation can compute the frequencies-of-occurrences by directly counting the frequencies of discretized distances and establishing a Huffman tree for each instance.

After the Huffman trees are established, this dissertation denotes the encoding length of instance $\exists j \neq i$ by $L_{i,j}$ in Huffman tree $i$. Finally, the anomaly score of an instance $i$ is
computed by

\[ score_i = \log \left( \frac{\sum_{j=1}^{n} L_{i,j}}{\sum_{j=1}^{n} L_{j,i}} \right), \forall 1 \leq i \leq n \] (2.4)

In Eqn. (2.4), the nominator represents the global information that the abnormality level is high if an instance is remote from other instances and the denominator represents the local information that the abnormality level is high if its distances from other instances are all located in high-frequency range and resulting in relatively short encoding lengths. Given the dataset, the end-to-end procedure for computing the anomaly scores of all instances is depicted in Algorithm 1. Note that the distances between all instance pairs need to be computed in order to build the Huffman forest, and hence, the time complexity of the proposed method is \( O(n^2) \). This will limit the application of the proposed method on very large datasets. To tackle this issue, a sampling-based version of HuffForest is also presented in this work.

Algorithm 1: HuffForest Anomaly Detection

**Input**: \( \{x_i\}_{1 \leq i \leq n} \)

**Output**: \( score_i \forall 1 \leq i \leq n \)

1. Normalize all dimensions for each instance \( x_i \in \{x_i\}_{1 \leq i \leq n} \) according to Eqn. (2.1).

2. Compute the distance between each pair of instances \( i \) and \( j \) as \( d_{ij} \) according to Eqn. (2.2).

3. For each instance \( i \), compute the normalized distances \( d_i^*(x_j) \) according to Eqn. (2.3).

4. For each instance \( i \), build the Huffman tree for the rest instances with frequencies-of-occurrence created by counting \( d_i^*(x_j), \forall j \neq i \).

5. Denote the encoding length of instance \( j \exists j \neq i \) by \( L_{i,j} \) in Huffman tree \( i \).

6. Compute the abnormality level of each instance \( i \) as \( score_i \) according to Eqn. (2.4).

7. return \( score_i \forall 1 \leq i \leq n \)
2.2 Sampling-Based HuffForest

In order to overcome the difficulties induced by the high computational complexity of HuffForest, a sampling-based version is further developed to identify anomalies based on a subset of the original dataset. While sub-sampling can substantially reduce the time complexity, it does not affect the capability for anomaly detection if the sampling does not change the data distribution. Furthermore, it is also proven to be an effective approach to tackle masking and swamping effects [31, 33, 40]. While masking and swamping are common problems of anomaly detection induced by the increasing of normal and abnormal instances [31], sub-sampling can reduce the number of instances considered for anomaly detection, thereby mitigating the masking and swamping effects.

In order to design an effective sampling mechanism, a critical issue is to preserve the distribution of the original dataset. Although a naive strategy can be uniformly sampling based on the spatial distribution of the instances, it is sensitive to the variation of density. Thus, this dissertation develops a novel distribution-preserving sampling strategy to accelerate the proposed HuffForest method while maintaining detection accuracy. Given a dataset, it first computes the geometric center of all instances, denoted by

\[
\overrightarrow{O} = \left( \frac{\sum_{i}^{n} x_{i1}}{n}, \frac{\sum_{i}^{n} x_{i2}}{n}, \ldots, \frac{\sum_{i}^{n} x_{im}}{n} \right),
\]

where \( x_{ik} \) is the \( k \)-th attribution of instance \( x_i \). Subsequently, the distance from each instance \( i \) to \( \overrightarrow{O} \) is computed as

\[
d_o(i) = \| x_i - \overrightarrow{O} \|_2, \forall 1 \leq i \leq n
\]

The instances are then sorted according to \( d_o(i) \) and uniformly arranged into \( B \) bins. Subsequently, \( \psi \) instances are sampled uniformly from the bins to construct a subset for identifying abnormal instances. In order to detect all the potential abnormal instances, the dataset is repeatedly sampled without putting back until no instance is left. Since a subset needs a sufficient number of instances to represent the distribution, if the number of instances in the
dataset is less than $2 \cdot \psi$ after certain sampling steps, all of them are selected in the last step to avoid the bias induced by insufficient sub-sampling. Given the dataset, the end-to-end steps of sampling-based HuffForest are depicted by Algorithm 2. Since the Huffman forest-based anomaly detection is applied on each subset, the time complexity is reduced to $O\left(\frac{n}{\psi} \psi^2\right) = O(n\psi)$.

**Algorithm 2: Sampling-Based HuffForest**

**Input**: $\{x_i\}_{1 \leq i \leq n}$

**Output**: $score\forall 1 \leq i \leq n$

1. Compute the Geometric centers of the dataset as $\vec{O}$ according to Eqn. (2.5).

2. For instance $i \in [1, n]$, compute the distance to the geometric center as $d_o(i)$ according to Eqn. (2.6).

3. Rank the instances according to $d_o(i)$ and uniformly arrange them into $B$ bins.

4. Initialize the number of un-sampled instances as $n_u = n$.

5. Initialize set $X_s = \{x_i\}_{1 \leq i \leq n}$

while $n_u \geq 0$ do
  $T \leftarrow \emptyset$;
  if $|X_s| \geq \psi$ then
    $T \leftarrow$ Uniformly sample $\psi$ instances from the $B$ bins;
    $X_s \leftarrow X_s \setminus T$
  else
    $T \leftarrow X_s$
  end
  Call Algorithm 1 to compute $score\forall i \in T$
end

return $score\forall i \in T$
2.3 Experimental Results

In this section, the proposed HuffForest method is evaluated on detection accuracy and time efficiency. Two existing techniques, iForest and LOF are used as baselines to demonstrate the advantages of the proposed methods. The experiments are conducted on a 2.10 GHz Linux cluster with 512 GB RAM. This dissertation implements HuffForest using Python 3.6 and directly call “scikit-learn” package for iForest and LOF. The detection accuracy and time efficiency are evaluated by AUC and time of computation, respectively.

The HuffForest method is evaluated and compared to baseline methods using two sets of experiments. In the first set of experiments, the aforementioned methods are analyzed in detail on 4 synthetic datasets to demonstrate the capability of the proposed method in tackling difficult cases. In the second set of experiments, the aforementioned methods are compared on a variety of public benchmarks to demonstrate the advantage of the proposed method on detection accuracy.

2.3.1 Performance on Synthetic Data

In this set of experiments, with prior knowledge of railroad structure, 4 synthetic datasets are generated to represent three scenarios mentioned in Chapter 1 Section 1.1 that contain the types of anomalies difficult for existing techniques. Figure 2.1 shows 4 synthetic datasets that represent local clustered anomalies, anomalies parallel to the axis, and anomalies surrounded by normal instances. Figure 2.1a represents the local clustered anomaly scenario alike Scenario I. Figure 2.1b and 2.1c are to evaluate the capabilities of detecting anomalies parallel to the axis which is the scenario similar to Scenario II. Figure 2.1d demonstrates the case of anomalies surrounded by normal instances referring to the scenario in Scenario III. The detection results are shown in Figure 2.1, and Table 2.1, Table 2.2, Table 2.3 and Table 2.4, respectively. In terms of LOF, two different search ranges ($k = 10$ and $k = 15$) are considered for a fair comparison.
(a) Dataset with locally clustered anomalies and local scattered anomalies.

(b) Dataset with local anomalies paralleled to the axis.

(c) Dataset with global anomalies paralleled to the axis.

(d) Dataset with anomalies surrounded by normal instances.

Figure 2.1: 4 synthetic datasets

Table 2.1: Local clustered experiment result

<table>
<thead>
<tr>
<th>Ranking</th>
<th>$x_c$</th>
<th>$x_g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>huffForest</td>
<td>1-7</td>
<td>8</td>
</tr>
<tr>
<td>iForest</td>
<td>10,105,12,27,29-32</td>
<td>1</td>
</tr>
<tr>
<td>LOF($k=10$)</td>
<td>41,47,52,56,65,67,82</td>
<td>1</td>
</tr>
<tr>
<td>LOF($k=15$)</td>
<td>42,45-47,54,57,59</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 2.1a and Table 2.1 show a dataset with 7 local clustered and 1 local scattered anomaly. In this example, 308 instances are generated including 8 anomalies (7 local clustered anomalies and 1 local scattered anomaly). Among the three detection methods, the AUC of HuffForest is 1, which means that it ranked all anomalies at the top. As shown in the experimental results, the anomalies are ranked in the top 8 by the Huffman forest. On the contrary, the baseline methods can only detect the scattered local anomaly and none of them can correctly detect the clustered local anomalies.

Table 2.2: Local anomalies parallel to the axis experiment result

<table>
<thead>
<tr>
<th>Ranking</th>
<th>$x_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>huffForest</td>
<td>1-6</td>
</tr>
<tr>
<td>iForest</td>
<td>7,11,13,18,19,25</td>
</tr>
<tr>
<td>LOF ($k=10$)</td>
<td>1-6</td>
</tr>
<tr>
<td>LOF ($k=15$)</td>
<td>1-6</td>
</tr>
</tbody>
</table>

Table 2.3: Global anomalies parallel to the axis experiment result

<table>
<thead>
<tr>
<th>Ranking</th>
<th>$x_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>huffForest</td>
<td>1-13</td>
</tr>
<tr>
<td>iForest</td>
<td>2,10,13,14,18,20,22,26,27,31,35,37,39</td>
</tr>
<tr>
<td>LOF ($k=10$)</td>
<td>20,92,113,176,181,206,227,234,272,283,286,287,310</td>
</tr>
<tr>
<td>LOF ($k=15$)</td>
<td>49,57,72-82</td>
</tr>
</tbody>
</table>

In order to demonstrate the capability of the Huffman forest on the detection of anomalies paralleled to the axis, two datasets are generated as shown in Figure 2.1b and Figure 2.1c. Figure 2.1b depicts a dataset consisting of 82 instances with 6 anomalies located between two normal instance clusters. As shown in the example of Table 2.2, 6 anomalies are generated.
Table 2.4: Anomalies surrounded by normal instance experiment result

<table>
<thead>
<tr>
<th>Ranking</th>
<th>$x_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>huffForest</td>
<td>1-11</td>
</tr>
<tr>
<td>iForest</td>
<td>125,147,148,151,159,164,170,187,197,199,207</td>
</tr>
<tr>
<td>LOF ($k=10$)</td>
<td>86,135,141,161,171,173,183,184,233,267,268</td>
</tr>
<tr>
<td>LOF ($k=15$)</td>
<td>1-11</td>
</tr>
</tbody>
</table>

between the normal clusters in parallel with the y-axis. HuffForest and LOF (with both $k = 10$ and $k = 15$) can rank the anomalies at the top of the list. However, iForest cannot make an accurate detection. Huffman forest and LOF with both configurations can rank the anomalies at the top of the anomaly list while the abnormal instances are ranked much lower in the list of iForest.

Figure 2.1c depicts a dataset of 313 instances with 13 anomalies remote from the main cluster of normal instances. As shown in the result in Table 2.3, 13 anomalies are generated parallel in parallel with y-axis. HuffForest is the only method that ranks the anomalies at the top of the list. the abnormal instances are ranked at the top in the list of Huffman forest while they are ranked much lower in the list of other approaches.

In Figure 2.1d, a dataset consisting of 311 instances including 11 anomalies surrounded by the normal instances. In this example, the normal instances form a ring, and 11 anomalies are surrounded by this ring. As shown in the detection results, both Huffman forest and LoF with $k = 15$ can rank the abnormal instances at the top of anomaly lists while other methods cannot generate an accurate detection result. Overall, the Huffman forest demonstrates a superior detection capability on all the considered difficult scenarios while none of the baseline methods can tackle them simultaneously.
2.3.2 Performance on Public Benchmarks

Although the proposed method outperforms on datasets synthesized according to railroad knowledge, evidence of advances will be strengthened if it shows superiority in a wider range of anomaly detection. Thus, the HuffForest method is further evaluated on the public benchmarks that are widely adopted in the domain of anomaly detection in this part. The original version of HuffForest is compared to iForest and LOF on 12 benchmarks as shown in Table 2.5, where the best detection accuracies (measured by AUC) are emphasized with bold symbols. In this set of experiments, HuffForest (HF) has no hyper-parameter to tune. For iForest (iF), this dissertation keeps the default configuration such that the iForest consists of iTrees while each iTree is built using 256 attributes. In order to guarantee the fairness of
Table 2.6: Comparison of S-HF, iF and LOF on 3 large benchmark datasets

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>d</th>
<th>Anomalies</th>
<th>AUC</th>
<th>Time (Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S-HF</td>
<td>iF</td>
<td>LOF</td>
<td>S-HF</td>
<td>iF</td>
</tr>
<tr>
<td>Http</td>
<td>567,479</td>
<td>3</td>
<td>2,211(0.4%)</td>
<td>1.00</td>
<td>0.97</td>
</tr>
<tr>
<td>F-Cover</td>
<td>286,048</td>
<td>10</td>
<td>2,747(0.9%)</td>
<td>0.91</td>
<td>0.88</td>
</tr>
<tr>
<td>Mulcross</td>
<td>262,144</td>
<td>4</td>
<td>26,214(10%)</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

comparison, this dissertation runs iForest 30 times on each dataset, and the trial with the largest AUC is picked. For LOF, this dissertation fixes the search range as $k = 10$. This dissertation can observe from the table that only on the Musk dataset, the Huffman forest (AUC=0.99) is slightly inferior to iForest (AUC=1). On the rest 11 benchmarks, it leads the detection accuracy. Although the time for computation is longer than the baseline methods, it is still within an acceptable range.

The sampling-based version of HuffForest is evaluated on 3 large benchmarks with over 200,000 instances. This dissertation selects the sub-sampling size as $\psi = 150$ and the number of bins as $B = 10$ for the sampling-based HuffForest and keeps the configurations of baseline methods the same as the last set of experiments. The results are shown in Table 2.6, where the best detection accuracies (measured by AUC) are emphasized with bold symbols. S-HF refers to sampling-based HuffForest. This dissertation can observe from the table that sampling-based HuffForest outperforms the baseline methods on all these benchmarks. Although it needs a longer computational time, the sampling-based Huffman forest offers a reasonable choice for applications where detection accuracy is more critical, which is usually the case for railroad station operations optimization.
Chapter 3: Optimality Analysis of Train Platforming and Routing with Different Interlocking Modes

3.1 Problem Formulation

Figure 3.1: Railroad station example

Generally, a passenger railway station (PRS) consists of a set of tracks $E$ and a set of nodes $V$, including endpoints and switches. Tracks are connections between two nodes $E \subseteq \{\{v_i, v_j\}|i, j \in V\}$ and each track $e_{v_i, v_j} \in E$ has a length $l_{v_i, v_j}$ from node $v_i$ to node $v_j$. Figure 3.1 presents the logic structure of a station, where switches are marked with three digits numbers and end points are marked with letter-number. All notation are presented in table.

---

2This chapter is derived in part from an article published in Communications in Transportation Research [62]
Table 3.1: Table of notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>The set of all nodes in a station</td>
</tr>
<tr>
<td>( v_i, v_j )</td>
<td>Nodes ( v_i ) and ( v_j ) from ( V )</td>
</tr>
<tr>
<td>( E )</td>
<td>The set of all tracks in a station</td>
</tr>
<tr>
<td>( e_{v_i,v_j} )</td>
<td>The track connecting nodes ( v_i ) and ( v_j ), ( e_{v_i,v_j} \in E )</td>
</tr>
<tr>
<td>( l_{v_i,v_j} )</td>
<td>The length of track ( e_{v_i,v_j} )</td>
</tr>
<tr>
<td>( F )</td>
<td>The set of all directions</td>
</tr>
<tr>
<td>( f_v )</td>
<td>The direction endpoint ( v ) is associated to, ( v \in V, f_v \in F )</td>
</tr>
<tr>
<td>( P )</td>
<td>The set of all platforms, ( P \subseteq E )</td>
</tr>
<tr>
<td>( p_v )</td>
<td>The platform endpoint ( v ) is associated to, ( v \in V, p_v \in P )</td>
</tr>
<tr>
<td>( R )</td>
<td>The set of all routes</td>
</tr>
<tr>
<td>( V_r )</td>
<td>The ordered set of nodes representing an unique route ( r ), ( r \in R )</td>
</tr>
<tr>
<td>( f_r )</td>
<td>The direction ( r ) connecting from (to), ( f_r \in F )</td>
</tr>
<tr>
<td>( p_r )</td>
<td>The platform ( r ) connecting from (to), ( p_r \in P )</td>
</tr>
<tr>
<td>( R_r^p )</td>
<td>The set of all routes connecting ( f ) and ( p ), ( R_r^p \subseteq R )</td>
</tr>
<tr>
<td>( K )</td>
<td>The set of all discretized time dimension</td>
</tr>
<tr>
<td>( T )</td>
<td>The set of trains to be routed to a platform</td>
</tr>
<tr>
<td>( f_t^A, f_t^D )</td>
<td>The arrival and departure direction of train ( t ), ( f_t^A, f_t^D \in F )</td>
</tr>
<tr>
<td>( \pi_t^A, \pi_t^D )</td>
<td>The scheduled arrival and departure time of train ( t ), ( \pi_t^A, \pi_t^D \in K )</td>
</tr>
<tr>
<td>( D_t )</td>
<td>All possible decision for train ( t )</td>
</tr>
<tr>
<td>( d_t )</td>
<td>One possible decision for train ( t ), ( d_t \in D_t )</td>
</tr>
<tr>
<td>( r_{A_d t}, r_{D_d t}^D )</td>
<td>Decided arrival and departure schedule for train ( t )</td>
</tr>
<tr>
<td>( r_{A_d t}, r_{D_d t}^D )</td>
<td>Decided arrival and departure route for train ( t )</td>
</tr>
<tr>
<td>( p_{d_t} )</td>
<td>Decided platform choice for train ( t )</td>
</tr>
<tr>
<td>( b_p )</td>
<td>Safety buffer on platform ( p )</td>
</tr>
<tr>
<td>( K_{p_{d_t}} )</td>
<td>The set of all time points decision ( d_t ) occupying platform ( p_{d_t} )</td>
</tr>
<tr>
<td>( v_{r,i} )</td>
<td>The ( i )-th switch route ( r ) uses</td>
</tr>
<tr>
<td>( l_{r,i} )</td>
<td>The track length from switch ( i-1 ) to switch ( i ) in a route, where ( l_{r,1} = 0 )</td>
</tr>
<tr>
<td>( b_s )</td>
<td>Safety buffer on switch ( s )</td>
</tr>
<tr>
<td>( K_{v_{d_t}}^{A}, K_{v_{d_t}}^{D} )</td>
<td>The set of all time points decision ( d_t ) occupying switch ( v ) by ( r_{d_t}^A ) and ( r_{d_t}^D ).</td>
</tr>
<tr>
<td>( K_{v_{d_t}} )</td>
<td>( K_{v_{d_t}} = K_{v_{d_t}}^{A} \cap K_{v_{d_t}}^{D} )</td>
</tr>
<tr>
<td>( d_t' )</td>
<td>Dummy decision for train ( t )</td>
</tr>
<tr>
<td>( x_{d_t} )</td>
<td>Decision variable for ( d_t )</td>
</tr>
<tr>
<td>( c_{d_t} )</td>
<td>Cost for decision ( d_t )</td>
</tr>
</tbody>
</table>
There are two types of endpoints: station endpoints and platform endpoints. Each station endpoint is associated with one arrival or one departure direction \( f_r \in F \), where \( F \) is the set of all directions. And each platform endpoint is associated with one platform \( p_r \in P \), where \( P \subset E \) is a set of tracks for trains to stop and perform boarding or alighting operations. For example in Figure 3.1, station endpoints I3, I4 are associated with the arrival direction from Zhengzhou, O1, O3 are associated with the departure direction to Chengdu. Platform endpoints L3 and R3 are associated with the platform (L3, R3).

The paths between station endpoints and platform endpoints are called routes in our context. Let \( R \) be the set of all routes, including inbounding routes going from arrival directions to platforms, and outbounding routes leaving from platforms to departure directions. Without loss of generality, this dissertation can use a collection of ordered \( V_r = \{ [v_i]_{i=1,...,|v_i|} | v_i \in V \} \) nodes to identify an unique route \( r \in R \). For instance, shown with red lines in figure 3.1, the set [I1,114,118,120,128,138,150,170,L12] can represent an inbounding route from I1 to platform (L12,R12).

Obviously, one route \( r \) contains only one station endpoint and platform endpoint, which is mapped to only one direction \( f_r \in F \) and only one platform \( p_r \in P \) respectively. This dissertation can obtain the set of all possible routes connecting direction \( f \) to platform \( p \) (or \( p \) to \( f \)) as \( R_{f}^{p} = \{ r | f_r = f, p_r = p, r \in R \} \) for each \( f \in F \) and \( p \in P \). Since the station endpoints are usually also the endpoints of the track circuit that is used to control safety, this dissertation regards all endpoints as special switches in the following contents.

Let \( \mathcal{K} = \{ k | k = 1, 2, ..., K \} \) be the set of all discretized time points, where \( K \) is the planning cycle, \( T \) be the set of trains to be routed to a platform within the planning. For each \( t \in T \), the predetermined information includes: a arrival direction \( f_t^A \in F \), a scheduled arrival time \( \pi_t^A \in \mathcal{K} \), a departure direction \( f_t^A \in F \), and a scheduled departure time \( \pi_t^D \in \mathcal{K} \).
3.1.1 TPR Decisions

Let $d_t$ be one possible decision for train $t \in T$, basically, without safety constraints, $d_t$ includes three sub-decisions:

- **Train Plaforming** decision, which is a feasible platform choice $p_{d_t} \in P$ satisfying both $|R^{p_{d_t}}_t| >= 1$ and $|R^{p_{d_t}}_t| >= 1$

- **Train Routing** decision, including one feasible inbounding route $r^A_{{d_t}} \in R^{p_{d_t}}_t$ and one feasible outbounding route $r^D_{{d_t}} \in R^{p_{d_t}}_t$.

- **Train Schedule Adjusting** decision, which is to determine the adjusted arrival time $\tau^A_{{d_t}} \in \{ k | \pi^A_t - \delta^A_t <= k <= \pi^A_t + \delta^A_t, \forall k \in K \}$ and the adjusted departure time $\tau^D_{{d_t}} \in \{ k | \pi^D_t - \delta^D_t <= k <= \pi^D_t + \delta^D_t, \forall k \in K \}$, where $\delta^A_t$ and $\delta^D_t$ is the maximum allowed time deviation from the original arrival and departure schedule. In addition, the adjusted time should satisfy $\tau^A_{{d_t}} - \tau^D_{{d_t}} >= \epsilon_t$, where $\epsilon_t$ is the a minimal dwell time for each train $t$.

By enumerating above decisions, the set of all feasible decisions $D_t$ for each train $t$ can be easily achieved and $d_t$ can be uniquely represented as $d_t = (\tau^A_{{d_t}}, r^A_{{d_t}}, p_{d_t}, r^D_{{d_t}}, \tau^D_{{d_t}}) \in D_t$. One decision example is illustrated in figure 3.1 with blue lines for Train G9527, where the predetermined arrival direction is Zhengzhou, departure direction is Chengdu, and original scheduled arrival and departure time is 8:50 am and 9:30 am respectively. This decision selects platform 6G, route [I3,109,111,123,127,141,159,6G] for inbounding, route [6G,166,152,144,134,126,104,03] for outbounding and adjusts the arrival and departure time to 8:48 am and 9:32 am.

3.1.2 Resource Occupations with Different Interlocking Modes

According to platform safety rule, each decision occupies the platform from $\tau^A_{{d_t}}$ to $\tau^D_{{d_t}} + b_p$, where $b_p$ is a safety buffer time interval on platform. Let $\mathcal{K}_{p_{d_t}} = \{ k | \tau^A_{{d_t}} <= k < \tau^D_{{d_t}} + b_p, \forall k \in \mathcal{K} \}$ be the set of all time points decision $d_t$ occupying platform $p_{d_t}$. 
According to the locking and release patterns for three interlocking modes in Chapter 1 Section 1.2, let $v_{r,i}$ be the $i$-th switch route $r$ uses, and $l_{r,i} = l_{r,i-1,v_{r,i}}$ for each $i = 1, ..., |V_r|$ represents the track length from switch $i - 1$ to switch $i$ in a route, where $l_{r,1} = 0$ is defined for a better demonstration purpose. Let inbounding(outbounding) speed be $s_t^A(s_t^D)$ and the train length be $l_t$.

Let $b_s$ be the safety buffer time interval on switch, and the $K_{v_{dt}}^A$ and $K_{v_{dt}}^D$ be the set of all time points decision $d_t$ occupying switch $v$ by $r_{dt}^A$ and $r_{dt}^D$ respectively. The $K_{v_{dt}}^A$ and $K_{v_{dt}}^D$, under different interlocking modes can be calculated follows:

1. RLRR:

$$K_{v_{dt}}^A = \left\{ k \left| \tau_{dt}^A \leq k < \tau_{dt}^A + \frac{\sum_{i=1}^{|V_r|} l_{r,i} + l_t}{s_t^A} + b_s, r = r_{dt}^A, k \in \mathcal{K} \right. \right\}, \forall v \in V_{v_{dt}}^A \quad (3.1)$$

$$K_{v_{dt}}^D = \left\{ k \left| \tau_{dt}^D \leq k < \tau_{dt}^D + \frac{\sum_{i=1}^{|V_r|} l_{r,i} + l_t}{s_t^D} + b_s, r = r_{dt}^D, k \in \mathcal{K} \right. \right\}, \forall v \in V_{v_{dt}}^D \quad (3.2)$$

where $\sum_{i=1}^{\sum_{i=1}^{l_t}} l_{r,i}$ is actually the length of route $r$.

2. RLSR:

$$K_{v_{dt}}^A = \left\{ k \left| \tau_{dt}^A \leq k < \tau_{dt}^A + \frac{l_{r,v}}{s_t^A} + b_s, r = r_{dt}^A, k \in \mathcal{K} \right. \right\}, \forall v \in V_{v_{dt}}^A \quad (3.3)$$

$$K_{v_{dt}}^D = \left\{ k \left| \tau_{dt}^D \leq k < \tau_{dt}^D + \frac{l_{r,v}}{s_t^D} + b_s, r = r_{dt}^D, k \in \mathcal{K} \right. \right\}, \forall v \in V_{v_{dt}}^D \quad (3.4)$$

where $\sum_{i=1}^{\sum_{i=1}^{l_t}} l_{r,i}$ is actually the sectional length from the first node to $v$ in route $r$. 


3. SLSR:

\[ \mathcal{K}_v = \left\{ k \left| \tau^A_{v} + \sum_{i=1}^{v_{i,v}} l_{r,i} \leq k < \tau^A_{v} + \sum_{i=1}^{v_{i,v}} l_{r,i} + l_t \right| + b_v, r = r^A_{v}, k \in \mathcal{K} \right\}, \forall v \in V_{r^A} \quad (3.5) \]

\[ \mathcal{K}_v = \left\{ k \left| \tau^D_{v} + \sum_{i=1}^{v_{i,v}} l_{r,i} \leq k < \tau^D_{v} + \sum_{i=1}^{v_{i,v}} l_{r,i} + l_t \right| + b_v, r = r^D_{v}, k \in \mathcal{K} \right\}, \forall v \in V_{r^D} \quad (3.6) \]

For any two decisions \( d^1_t \) and \( d^2_t \), they are conflicted on platform if \( \mathcal{K}_{p^1_d} \cap \mathcal{K}_{p^2_d} \neq \emptyset \). And let \( \mathcal{K}_{v_d} = \mathcal{K}_v \cap \mathcal{K}_{v_d} \) be the set of all time points decision \( d_t \) occupying switch \( v \). they are conflicted on route if \( \mathcal{K}_{v^1_d} \cap \mathcal{K}_{v^2_d} \neq \emptyset \). Beside, an dummy decision \( d'_t \) is added to \( D_t \) for each \( t \in T \) to represent the case of no feasible decision, where \( \mathcal{K}_{p^1_d} = \emptyset \) and \( \mathcal{K}_{v^1_d} = \emptyset \).

3.1.3 Space-time Formulation

Each decision is associated with a binary decision variable and a penalty. Let \( x_{d_t} \) be the binary decision variable for each \( d_t \in D_t \) and \( t \in T \) that \( x_{d_t} = 1 \) if \( d_t \) is selected, otherwise \( x_{d_t} = 0 \). Also let \( c_{d_t} \) be the penalty, which is usually determined by decision and train features in practice. For example, the dummy decisions are associated with comparative high penalties since all trains are expected to be assigned, time-adjusted decisions are associated with medium penalties based on the level of adjustment, and the non-preferred platform decisions are associated with a low penalty. The model formulation is as follows:
The model formulation is as follows:

\[
\begin{align*}
\text{min} & \quad \sum_{t \in T} \sum_{d_t \in D_t} c_{d_t} x_{d_t} \\
\text{s.t.} & \quad \sum_{t \in T} x_{d_t} = 1, \quad \forall t \in T \tag{3.8} \\
& \quad \sum_{t \in T} \sum_{d_t \in D_t} x_{d_t} \leq 1, \quad \forall k \in K, p \in P \tag{3.9} \\
& \quad \sum_{t \in T} \sum_{d_t \in D_t} x_{d_t} \leq 1, \quad \forall k \in K, v \in V \tag{3.10}
\end{align*}
\]

Constraints (3.8) impose that each train has to select only one decision. Constraints (3.9) and (3.10) are platform and route conflict constraints to ensure each space platform \( p \) or switch \( v \) can at most be occupied by one decision at each time \( k \). The advantages of space-time formulation include two aspects: (1) instead of exponential growing mutual exclusive constraints, this formulation sets up an upper bound of the number of constraints (3.9) and (3.10), which equals \((|P| + |V|) \ast |K|\) (2) this formulation doesn’t require detecting pairwise compatibility of decisions before the model is built, which saves a lot of computations in data preparation.

To better illustrate Constraints (3.9) and (3.10), let \( y^{k,p} \) and \( z^{k,v} \) be the space-time states (occupied or not) of \((p, k)\) and \((v, k)\) determined by \( x_{d_t} \). Figure 3.2 illustrated a simplified example for one decision under RLSR interlocking mode. Figure 3.2a described the decision and Figure 3.2b showed how space-time states are determined by decision \( x_{d_t} \), where each small cell represents the \( y^{k,p} \) or \( z^{k,p} \).
3.2 Numerical Experiments

3.2.1 Solution Approaches

To tackle the exponential growth complexity of the graph, the study [12] used clique inequalities to reduce solution time and also implemented a branch-and-cut-price algorithm to solve the model above. However, thanks to the development of cutting-edge solvers such as Gurobi or Complex, these clique search, redundant variables, and constraints detection, branching, and pricing techniques have already been handled inside the solver with excellent efficiency. This dissertation compared both methods that the state-of-the-art commercial
solvers outperform the proposed approaches. Thus, this dissertation used Gurobi 9.0 to solve the problem in the following experiments.

3.2.2 Experiment Inputs

3.2.2.1 Stations and Traffic Densities

The interlocking modes will directly affect the route conflict constraints and further contribute to the optimality of TPR. Intuitively, conducting TPR for larger stations is harder than that for smaller ones, where the size is comprehensively measured according to the number of directions, switches, tracks, platforms, and routes. To evaluate the interlocking optimality gaps under different station sizes, this work selected three real-world stations with large, medium, and small sizes respectively. The station profiles are shown in 3.2.

Table 3.2: Station profiles

<table>
<thead>
<tr>
<th>Station Name</th>
<th>Size</th>
<th>Directions</th>
<th>Switches</th>
<th>Tracks</th>
<th>Platforms</th>
<th>Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kunming</td>
<td>S</td>
<td>3</td>
<td>68</td>
<td>186</td>
<td>11</td>
<td>424</td>
</tr>
<tr>
<td>Shenzhenbei</td>
<td>M</td>
<td>4</td>
<td>82</td>
<td>192</td>
<td>20</td>
<td>641</td>
</tr>
<tr>
<td>Guangzhounan</td>
<td>L</td>
<td>7</td>
<td>112</td>
<td>275</td>
<td>28</td>
<td>778</td>
</tr>
</tbody>
</table>

As mentioned in Chapter 1 Section 1.2, the optimalities of TPR are significantly affected by traffic patterns and densities of train schedules, since the TPR planning is conducted after the network-wise train schedule is determined. For instance, determining the TPR of 10 trains within 6 hours is much easier than that within 1 hour. Thus, this dissertation should evaluate the optimality gaps of three interlocking modes under different traffic densities.

This dissertation uses minimum headway in directions to measure the traffic density. For example, an original TS with a minimum headway of 3 minutes on one arrival direction has a higher density than that with a minimum headway of 7 minutes. This work selected 3, 5, 7, 9, 11 minutes minimum headway to represent different density scenarios.
3.2.2.2 Train Schedules

Using real-world train schedules as inputs may not provide enough insights into the optimality sensitivities, since the current train schedule is the result of RLRR mode planning and is usually conservative in operation with a relatively low density.

Thus, this dissertation synthesized several input train schedules within certain time windows based on various station and headway settings. However, since the experimental goal is to analyze the impacts of interlocking modes which only affect the route conflict constraints, there is an additional requirement for the input train schedule - the TPR optimalities on input train schedules should be mostly impacted by route conflict constraints rather platform conflict ones.

To be more specific, given an input train schedule, if this dissertation relaxes the route conflict constraints and finds an optimal TPR with an objective value equal to zero, it can imply that platform conflict constraints are not hard ones for this problem and the full optimality will mostly be determined by route conflict constraints.

According to the above features, qualified input train schedules can be generated by following procedures and the input cases are shown in table 3.3:

1. Generate a few random trains and add them candidate pool, where their schedules are within the time window and satisfy station and traffic density (minimum headway) settings.

2. Conduct the modified TPR for candidate trains, where the route conflict constraints are relaxed and the penalties for dummy decisions are set to 1 (others are set to 0).

3. Remove trains with dummy decisions.

4. Repeat the above procedures until no more trains can be added to the candidate pool.
### Table 3.3: Input cases

<table>
<thead>
<tr>
<th>Station</th>
<th>Case ID</th>
<th>Time Horizon</th>
<th>Minimum Headway</th>
<th>Trains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kunming (KM)</td>
<td>1</td>
<td>240</td>
<td>3</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>300</td>
<td>5</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>420</td>
<td>7</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>540</td>
<td>9</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>600</td>
<td>11</td>
<td>87</td>
</tr>
<tr>
<td>Shenzhenbei (SZB)</td>
<td>6</td>
<td>210</td>
<td>3</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>300</td>
<td>5</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>450</td>
<td>7</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>540</td>
<td>9</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>600</td>
<td>11</td>
<td>132</td>
</tr>
<tr>
<td>Guangzhou (GZN)</td>
<td>11</td>
<td>180</td>
<td>3</td>
<td>148</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>210</td>
<td>5</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>270</td>
<td>7</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>300</td>
<td>9</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>420</td>
<td>11</td>
<td>144</td>
</tr>
</tbody>
</table>

#### 3.2.3 Experiment Result

In this section, the experiments are conducted and the optimalities of three interlocking modes are demonstrated. In each experiment, the time dimension is discretized in minutes, the speed limit on each route is used as the default train speed, the length of CRH380AL is used as the default train length, and the allowed time deviation for arrival and departure is set to 3 minutes. The penalty for decisions with an original train schedule is set to 0,
Table 3.4: Experiment results

<table>
<thead>
<tr>
<th>Station</th>
<th>Case ID</th>
<th>Headway</th>
<th>Trains</th>
<th>Objective Value</th>
<th>Dummy decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>RLRR</td>
<td>RLSR</td>
<td>SLSR</td>
</tr>
<tr>
<td>KM</td>
<td>1</td>
<td>3</td>
<td>88</td>
<td>223</td>
<td>174</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>83</td>
<td>78</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>84</td>
<td>55</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>9</td>
<td>86</td>
<td>25</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>11</td>
<td>87</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>SZB</td>
<td>6</td>
<td>3</td>
<td>135</td>
<td>294</td>
<td>264</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>5</td>
<td>132</td>
<td>81</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>7</td>
<td>133</td>
<td>38</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>131</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>132</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>GZN</td>
<td>11</td>
<td>3</td>
<td>148</td>
<td>274</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>5</td>
<td>140</td>
<td>118</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>7</td>
<td>145</td>
<td>65</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>9</td>
<td>139</td>
<td>63</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>11</td>
<td>144</td>
<td>28</td>
<td>10</td>
</tr>
</tbody>
</table>

whereas that with an adjusted one is set to 1. The penalty for dummy decisions is set to 10. The overall experiment results are presented in table 3.4.

3.2.3.1 Sensitivity Analysis on Traffic Densities

Figure 3.3 shows the optimalities of three interlocking modes under different headway settings for each station. First, this dissertation can observe a general trend that the objective value decreases significantly with the increase of the minimum headway. This fact is quite intuitive that a higher minimum headway means a lower traffic density and fewer potential
Figure 3.3: Optimalities on different minimum headways

train conflicts, then easier to find a better solution. If the headway is large enough, the
model will be unconstrained on train conflicts.

In addition, this dissertation can also observe that the performance of SLSR is more
stable than the other two modes, which implies, that in this mode, switch resources are less
likely to be bottlenecks when pursuing global optimalities under this mode.

More importantly, this dissertation can see that both RLSR and SLSR modes can bring
remarkable optimality improvements compared to the RLSR model, especially in higher
traffic density scenarios. Meanwhile, this dissertation can also see from table 3.4 that these
improvements are mostly credited to the reduction of dummy decisions, which means more
trains were platformed and infrastructures were better utilized.

From an engineering practical perspective, this dissertation could conclude that RLRR
mode is a cost-effective configuration for TPR planning in low-density scenarios. However,
in high-density scenarios, to take the best advantage of station capacities, this dissertation
should consider RLSR mode or even SLSR if the technology supports it.
3.2.3.2 Sensitivity Analysis on Station Sizes

This dissertation also analyzed the optimality improvements under different station sizes, which are measured by calculating the lift of each mode.

\[ \text{lift}_{\text{mode}} = \frac{|\text{obj}_{\text{mode}} - \text{obj}_{\text{base}}|}{\text{obj}_{\text{base}}} \]  

(3.11)

where the \( \text{obj}_{\text{mode}} \) is the objective value of target mode and \( \text{obj}_{\text{mode}} \) is the that of baseline mode - RLRR interlocking mode. Let \( \text{lift}_{\text{mode}} = 0 \) if \( \text{obj}_{\text{base}} = 0 \).

Since Section 3.2.3.1 has pointed out that the improvement is more significant for high-density scenarios, the station-size-wise comparisons are only conducted on high-density cases. Here this dissertation selects Case 1,2,6,7,11,12 from above and three additional cases as in table 3.5:

Table 3.5: Additional experiment results

<table>
<thead>
<tr>
<th>Station</th>
<th>Case ID</th>
<th>Headway</th>
<th>Trains</th>
<th>Objective Value</th>
<th>Dummy decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RLRR</td>
<td>RLSR</td>
</tr>
<tr>
<td>KM</td>
<td>16</td>
<td>4</td>
<td>86</td>
<td>161</td>
<td>106</td>
</tr>
<tr>
<td>SZB</td>
<td>17</td>
<td>4</td>
<td>132</td>
<td>191</td>
<td>104</td>
</tr>
<tr>
<td>GZN</td>
<td>18</td>
<td>4</td>
<td>142</td>
<td>138</td>
<td>72</td>
</tr>
</tbody>
</table>

Figure 3.4: Lifts on three stations
Figure 3.4 shows the lift of RLSR and SLSR modes at different station and headway settings. As this dissertation can see from the figure, the SLSR mode can always provide a good lift in spite of the station size. On the other hand, for the RLSR mode, the overall trend shows that larger stations tend to provide better lifts.

In practical words, although the implementation of sectional-locking sectional-route requires more advanced technologies that can handle precision control, the expected benefits of deploying this mode are really remarkable for all sizes of stations. If the technology and management are not ready enough, investing RLSR on large-size stations can also hopefully bring significant returns.
Chapter 4: An Automation Solution to Convert CAD Engineering Drawings into Railroad Station Models

4.1 Railroad Station Symbol Recognition

4.1.1 Basic Primitives and Operations

Basic primitives (BP) are fundamental elements forming symbols. In DXF files, there are limited types of basic primitives including points, lines, arcs, arrows, ellipses, etc. Specifically, in a railroad engineering drawing, there are only two types of BPs - lines and arcs. For engineering convenience, define $s$ be a unique BP, $s$ is defined as

$$s = (t, l, p, p_1, p_2, f)$$

where $t$ is the type of BP; $l$ is the length of BP; $p$ indicates the position of BP which is represented by the coordinate of the center point for a line and the center of the circle for an arc; $p_1$ and $p_2$ is the coordinate of two ends of a BP respectively; $f$ is a logical variable indicating whether an arc is inferior. Figures 4.1 show three examples of line and arc definition. Specifically, a point is a special type of line where $l = 0$. 

Figure 4.1: Basic primitive examples

Basic primitives (BP) are fundamental elements forming symbols. In DXF files, there are limited types of basic primitives including points, lines, arcs, arrows, ellipses, etc. Specifically, in a railroad engineering drawing, there are only two types of BPs - lines and arcs. For engineering convenience, define $s$ be a unique BP, $s$ is defined as

$$s = (t, l, p, p_1, p_2, f)$$

where $t$ is the type of BP; $l$ is the length of BP; $p$ indicates the position of BP which is represented by the coordinate of the center point for a line and the center of the circle for an arc; $p_1$ and $p_2$ is the coordinate of two ends of a BP respectively; $f$ is a logical variable indicating whether an arc is inferior. Figures 4.1 show three examples of line and arc definition. Specifically, a point is a special type of line where $l = 0$. 

49
Given a set of primitives, a derived set of basic primitives or values can be obtained via certain calculations. This process is called primitive operations in the most state-of-the-art CAD software. Specifically, it is called binary primitive operations if there are only 2 input basic primitives. Thanks to the sustainable development of CAD tools, many frequently used operations have been built within CAD software. Table 4.1 lists a few commonly used binary operation examples.

<table>
<thead>
<tr>
<th>Operations</th>
<th>L-BP</th>
<th>R-BP</th>
<th>Return</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>Line</td>
<td>Arc</td>
<td>Line</td>
<td>Finding the shortest line connecting one endpoint to one BP</td>
</tr>
<tr>
<td>Subtract</td>
<td>Line</td>
<td>Line</td>
<td>Line</td>
<td>Subtract the intersecting part of two BPs from L-BP</td>
</tr>
<tr>
<td>Angle</td>
<td>Line</td>
<td>Arc</td>
<td>Angle values</td>
<td>Calculating the angle value between L-BP (line) and tangent of R-BP (arc) at the intersecting point</td>
</tr>
</tbody>
</table>

In the case of railroad station symbol recognition, only distance and angle operations will be involved in our method, and each one will be introduced in the following context. Let $D$ and $A$ be the distance and angle operation respectively.

$$s' = D(s_1, s_2)$$  \hspace{1cm} (4.2)

$$\alpha = A(s_1, s_2)$$  \hspace{1cm} (4.3)

Give line or arc primitives $s_1$ and $s_2$, let $s'$ be the result of distance operations. $l'$ is the distance value. It can also be assumed that the two ends $p_1'$ and $p_2'$ can represent the point on $s_1$ and $s_2$ respectively. Figure 4.2 shows four examples of distance operations.

Angle operation is to calculate the relative angle of two primitives. For engineering convenience, this operation is only considered when $p_1' = da = p_2 = p_2''$ in this method. Figure 4.3 shows three types of angle operations.
4.1.2 Break and Cluster

In the railroad station engineering drawings, a switch is usually represented by the intersection of lines or arcs, and different branches connecting the intersection point belong to different tracks. Therefore, in order to properly identify the switch symbols, a basic primitive should be broken into several primitives if it is intersected with other primitives at endpoints.

For any two BPs $s_1$ and $s_2$, let $s'$ be the result of distance operation on them. $s_1$ and $s_2$ can be considered as intersected if $l^s \leq \epsilon$, where $\epsilon$ is a minimal threshold.
In this case, if \( p_1' \neq p_1^{s_1} \) and \( p_1' \neq p_2^{s_1} \), the basic primitive \( s_1 \) should be broken into two basic primitives \( s_{11} \) and \( s_{12} \), where \( p_1^{s_{11}} = p_1^{s_1} \) and \( p_2^{s_{11}} = p_1^{s'} \) for \( s_{11} \) and \( p_1^{s_{12}} = p_1^{s'} \) and \( p_2^{s_{12}} = p_2^{s_1} \) for \( s_{12} \). The same mechanism should also be applied to \( s_2 \). Figure 4.4 shows three examples of breaking primitives.

![Figure 4.4: Basic primitive break examples](image)

It should be noted that if two lines parallel and stay very close (smaller than \( \epsilon \)) to each other, there will be an infinite number of intersecting points. Therefore, these two lines should be first combined into one unique long line before executing breaking operations. It should also be noted that breaking operations should be executed several times until all basic primitives are broken properly.

In practice, some points in basic primitives are actually referred as to the same point but have different coordinates in DXF files. This fact is caused due to the limitation of the
manual drawing method and the computer precision. In this case, these points should be clustered into one point with a unique coordinate.

For any two points $p_1$ and $p_2$, let $x_{p_1}, y_{p_1}$ and $x_{p_2}, y_{p_2}$ be their coordinate respectively. $p_1$ and $p_2$ should be clustered into one point if

$$0 < (x_{p_1} - x_{p_2})^2 + (y_{p_1} - y_{p_2})^2 \leq \mu \epsilon \quad (4.4)$$

where $\epsilon$ is the minimal error threshold and $\mu$ is the scale multiplier. The clustered coordinate equals

$$\left(\frac{x_{p_1} + x_{p_2}}{2}, \frac{y_{p_1} + y_{p_2}}{2}\right) \quad (4.5)$$

After executing breaking and clustering operations, all basic primitives (lines and arcs) are expected to be connected precisely at and only at the endpoints. So far, the DXF file has been properly preprocessed and ready for recognizing railroad station symbols.

### 4.1.3 Switch Symbol Recognition

Let $S$ be the set of all basic primitives, $N = \{n|n = p_1^s, \forall s \in S\} \cup \{n|n = p_2^s, \forall s \in S\}$ be the set of all endpoints. For each $n \in N$, the set of basic primitives connected at $n$ can be obtained as

$$S_n = \{s|n = p_1^s, \forall s \in S\} \cup \{s|n = p_2^s, \forall s \in S\}, \forall n \in N \quad (4.6)$$
where $S_n \subset S$. Then, according to the number of basic primitives associated with $n$, $N$ can be divided into three subsets,

$$N_{end} = \{ n | |S_n| = 1, \forall n \in N \}$$ (4.7)

$$N_{con} = \{ n | |S_n| = 2, \forall n \in N \}$$ (4.8)

$$N_{sw} = \{ n | |S_n| > 2, \forall n \in N \}$$ (4.9)

$$N = N_{end} \cup N_{con} \cup N_{sw}$$ (4.10)

where $N_{end}$ is the set of all station entering or exiting points, $N_{con}$ is the set of all points connecting two primitives, and $N_{sw}$ is the set of all switches.

![Switch symbol examples](image)

Figure 4.5: Switch symbol examples

Figure 4.5 shows two typical switch symbols. It should be pointed out that train movements are subject to a minimal movement angle rule causing some movement directions to be invalid in practice. For example in Figure 4.5 (a), trains from $s_3$ can move to $s_1$ and $s_2$, but trains from $s_1$ is not valid to move to $s_2$, since the angle between $s_1$ and $s_2$ is too small for trains to move.

Hence, in addition, the valid direction should also be detected and recorded for each switch. For each $n \in N_{sw}$, let $D_n$ be set of all valid directions, $\theta$ be the minimal required angle. $D_n$ can be obtained as

$$D_n = \{ (s_1, s_2) | A(s_1, s_2) \geq \theta, s_1, s_2 \in S_n \}, \forall n \in N_{sw}$$ (4.11)
4.1.4 Track Symbol Recognition

After the switch symbols are recognized, it is straightforward to recognize track symbols. A switch symbol can be defined as a sequenced set of basic primitives where any consecutive two primitives are connected via a connection node \( n \in N_{con} \) and the start/end node is either an endpoint node or a switch node.

Let \( E \) be the set of all tracks. For each \( e \in E \), this dissertation first defines \( N_e \) be a sequenced set of nodes that can represent a unique track,

\[
N_e = \{[n_i]_{i=1,...,|n|}|n_1 \in N_{end} \cup N_{sw}, \quad n_i \in N_{con}, 1 < i < |n|, \\
|S_{n_i} \cap S_{n_{i+1}}| \geq 1, \\
1 < i < |n|, n_{|n|} \in N_{end} \cup N_{sw}, \}
\]

(4.12)

Then, let \( n_i \) be the \( i \)th element of \( N_e \), \( e \) can be defined as

\[
e = \{[s_j]_{j=1,...,|s|}|s_j \in S_{n_j} \cap S_{n_{j+1}}\} \forall e \in E
\]

(4.13)

In addition, a simple depth-first search (DFS) to recognize all tracks can be seen in Algorithm 3.

4.2 Railroad Station Modeling Automation Solution

In this section, this dissertation will introduce the proposed solution for converting recognized symbols to undirected railroad station models. And on top of the generated graph, the models will also cover inputs, required configuration, and route generation.
Algorithm 3: DFS to recognize all tracks

Input : $S$, $N_{con}$, $N_{sw}$, $N_{end}$
Output: $E$

1. $E \leftarrow \emptyset$
2. $V \leftarrow \emptyset$

3. for $n \in N_{sw} \cup N_{end}$ do

   4. for $s \in S_n \setminus V$ do

      5. $e \leftarrow \emptyset$
      6. $e \leftarrow e + [s]$, $V \leftarrow V + [s]$
      7. while $\exists \bar{s} \notin V$ and $\{\bar{s}, s\} = S_{\bar{n} \in N_{con}}$ do

         8. $e \leftarrow e + [\bar{s}]$, $V \leftarrow V + [\bar{s}]$
         9. $s \leftarrow \bar{s}$

   10. end

11. $E \leftarrow E + [e]$

12. end

13. end

4.2.1 Railroad Station Graph and Required Configurations

A railroad station graph model should first be generated using the recognized symbols. The modeling method is inspired by the reference [62]. More details can be found there.

Let $G = (T, V)$ be the station graph, where $T$ is the set of all tracks and $V$ is the set of all nodes including endpoints $V_{end}$ and switches $V_{sw}$ that $V = V_{sw} \cup V_{end}$. Tracks are connections between two nodes $T \subseteq \{(v_i, v_j) | i, j \in V\}$ and each track $t_{v_i, v_j} \in T$ has a length $l_{v_i, v_j}$ from node $v_i$ to node $v_j$.

Therefore, an unique $v \in V_{end}$ and $v \in V_{sw}$ can be created and mapped to one recognized $n \in N_{end}$ and $n \in N_{sw}$ respectively that $v \mapsto n$. In the same way, an unique track $t_{v_i, v_j} \in T$ can also be created and mapped to recognized $e \in E$ respective that $t_{v_i, v_j} \mapsto e$ if $v_i \mapsto n_{e, 1}^1$, $v_j \mapsto n_{e, 2}^1$. Subsequently, $l_{v_i, v_j}$ can be calculated as

$$l_{v_i, v_j} = \sum_{s \in t_{v_i, v_j} \mapsto e} l_s$$ \hspace{1cm} (4.14)

Let $T_{end} = \{t_{v_i, v_j} | v_i \in V_{end}, t_{v_i, v_j} \in T\} \cup \{t_{v_i, v_j} | v_j \in V_{end}, t_{v_i, v_j} \in T\} \subset T$ be the set of all end tracks, and $T_{sw} = \{t_{v_i, v_j} | v_i, v_j \in V_{sw}, t_{v_i, v_j} \in T\} \subset T$ be the set of all connection tracks.
After the railroad graph is created, some required configurations should be maintained by users manually. Let $F$ is the set of all directions, each station endpoint $v \in V_{\text{end}}$ is mapped to one arrival or departure direction $f_v \in F$. And because each endpoint $v_{\text{end}}$ is also mapped to one and only one track, without losing generality, each $t_{\text{end}} \in T_{\text{end}}$ can be assigned with a direction $f_t \in F$. Subsequently, let $T^0_{\text{end}}$ and $T^1_{\text{end}}$ be the set of all end tracks associated with entering and exiting directions respectively.

Figure 4.6a shows an example of manually configuring entering and exiting directions to end tracks.

Platform tracks are a subset of tracks for trains to park. Let $T_p \subset T$ be the set of all platform tracks. Each $t_p \in T_p$ should also be configured and assigned a name, which is usually numbered in practice. Figure 4.6b shows an example of manually configuring platform tracks.

### 4.2.2 Entering andExiting Routes Search

In a classical train platforming and routing problem, the paths connecting station endpoints and platforms are referred as to routes. In general, there are two types of routes - inbounding and outbounding, which arrives from endpoints and departures from platforms respectively. Let $R$ be set for all routes, a set of ordered tracks $T_r = \{(t_i)_{i \in 1,...,|t|} | t_i \in T\}$ can be used to identify an unique route $r \in R$.

Given $T_{\text{end}}$ and $T_p$, the next step is to enumerate all possible routes using $T_{\text{end}}$ and $T_p$. Recall the minimal movement angle rule introduced in 4.1, given any two tracks $t_1$ and $t_2$ connecting on switch $v_{\text{sw}} \in V_{\text{sw}}$, train movement is only allowed if they satisfy the minimal movement angle rule.

$$D_v = \{((t_1, t_2)|(s_1, s_2) \in D_n, v \mapsto n, s_1 \in t_1, s_2 \in t_2), \forall v \in V_{\text{sw}} \} \quad (4.15)$$
(a) End-track configuration example

(b) Platform configuration example

(c) Route configuration example

Figure 4.6: Configuration examples
For each \( t \in T \), a set of tracks \( T_t \) can be found that trains are allowed to move from \( t \) to \( t' \in T_t \). Let \( v_i, v_j \) and \( v'_i, v'_j \) be the nodes of \( t \) and \( t' \) respectively, let \( \{v_i, v_j\} \cup \{v'_i, v'_j\} = v_{t,t'} \in V_{sw} \) be their shared node.

\[
T_t = \{t' | (t, t') \in D_{v_{t,t'}}, t' \in T \}, \forall t \in T
\]  

(4.16)

Then, all possible routes can be enumerated using the steps in Algorithm 4. Figure 4.6c shows an example of generated routes.

---

**Algorithm 4: Routes generation**

Input : \( T, T_{end}^0, T_{end}^1, T_p \)  
Output: \( R \)

1. \( R \leftarrow [] \);  
2. for \( t \in T_{end}^0 \cup T_p \) do  
3. \( L \leftarrow [], G \leftarrow [] \);  
4. \( L \leftarrow L + [t] \);  
5. \( T_r \leftarrow [t], G \leftarrow G + [T_r] \);  
6. while \( |L| > 0 \) do  
7. \( t \leftarrow L.pop() \) — take out the last element in \( L \);  
8. \( T_r \leftarrow G.pop() \) — take out the last element in \( G \);  
9. if \( t \in T_p \) then  
10. \( R \leftarrow R + [T_r] \);  
11. break;  
12. end  
13. for \( t' \in T_t \) do  
14. if \( t' \not\in T_r \) then  
15. \( L \leftarrow L + [t'] \);  
16. \( T_r \leftarrow T_r + [t'] \);  
17. \( G \leftarrow G + [T_r] \);  
18. end  
19. end  
20. end
4.2.3 Summarized Procedures of Railroad Station Modeling

So far, the complete procedures of converting DXF files to undirected railroad station models have been finished. The full procedures are summarized in Figure 3.

Although this solution is not absolutely automatic and some required configurations need to be maintained manually, human interactions can be minimized to an acceptable degree with an excellent user-friendly tool. Thus, this work can still be considered a great supplement to railroad station modeling techniques.

In fact, in practice, some important information is included in DXF files as well, such as the directions, platform numbers, and signal locations. Recognizing this configuration information is another non-trivial task due to the complex symbol variants, and is not covered in this study.

4.3 Numerical Experiments

In the previous section, this dissertation has shown the complete procedures for converting DXF files into railroad station models. This solution has been developed into a user-friendly GUI application so that required information can be configured and maintained with minimal user interactions.

4.3.1 Experiment Goal and Data

The essential goal of this solution is to efficiently convert graphic basic primitives without explicit engineering interpretations into recognizable railroad symbols with obvious physical meanings. Thus, in order to demonstrate its sufficient engineering practicalities and values, this solution should be validated and measured from three aspects.

- **Validity.** The proposed solution should be able to generate valid results. To be specific, since the station model is the foundation of other production applications, the
engineering practice requires the recognizing and modeling results to be absolutely accurate and valid.

• **Productivity.** The proposed solution should be productive enough in assisting users in modeling railroad stations. This solution should be computationally efficient while the user interactions should be minimized to an acceptable level.

• **Applicability.** The outputs of the proposed solution can be friendly recognized and applied in other railroad station applications.
The experiments are conducted in four Class I real-world stations in Asia - Shenzhenbei Station (SZB), Guangzhounan Station (GZN), and Zhengzhoudong Station (ZZD), and Xianbei Station (XAB). An example of DXF files can be seen from Figure 1.5.

4.3.2 Validity Analysis

Table 4.2: Validity analysis on four DXF files

<table>
<thead>
<tr>
<th>Station</th>
<th>BPs</th>
<th>Switches</th>
<th>Tracks</th>
<th>Platforms</th>
<th>Routes</th>
<th>Enters</th>
<th>Exits</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SZB</td>
<td>343</td>
<td>192</td>
<td>82</td>
<td>20</td>
<td>641</td>
<td>4</td>
<td>4</td>
<td>100%</td>
</tr>
<tr>
<td>GZN</td>
<td>464</td>
<td>275</td>
<td>112</td>
<td>28</td>
<td>778</td>
<td>7</td>
<td>7</td>
<td>100%</td>
</tr>
<tr>
<td>ZZD</td>
<td>807</td>
<td>358</td>
<td>153</td>
<td>32</td>
<td>669</td>
<td>10</td>
<td>10</td>
<td>100%</td>
</tr>
<tr>
<td>XAB</td>
<td>440</td>
<td>265</td>
<td>123</td>
<td>27</td>
<td>485</td>
<td>5</td>
<td>5</td>
<td>100%</td>
</tr>
</tbody>
</table>

In the experiment, the validity is measured by testing if the proposed solution can recognize all symbols and generate all possible routes accurately on four DXF files. The output of each DXF file is compared with the actual station layout and operations. And the result in Table 4.2 shows that all DXF files can be accurately converted to station models.

It should be noted that all four DXF files are original files without any modifications. However, this work should admit that the excellent validity is based on an assumption that all four DXF are constructed under similar rules. For example, the crossing of basic primitives indicates there is a flyover from one to another unless the interaction point is very close to one of the endpoints.

4.3.3 Productivity Analysis

One of the important contributions of the proposed solution is expected to minimize manual work for modeling railroad stations. To evaluate the productivity benefits, this work invited a railroad engineering expert to manually recognize symbols and build station models.

For a fair comparison, all basic primitives in a DXF file are labeled with a unique ID. The human expert is required to provide the following information: (1) a list of all tracks, where
each track is represented by a list of basic primitive IDs; (2) a list of switches, where each
switch is associated with a coordinate and contains a list of tracks. And a list of allowed
directions from track to track should also be provided; (3) a list of routes, where each route
consists of a list of tracks. Be aware that, the output of the human expert is required to be
100% valid as well.

Table 4.3: Productivity analysis on four DXF files

<table>
<thead>
<tr>
<th>Station</th>
<th>Break</th>
<th>Cluster</th>
<th>Generate Graph</th>
<th>Configure Directions</th>
<th>Generate Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SZB</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>15 min</td>
<td>&lt;4s</td>
</tr>
<tr>
<td>GZN</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>28 min</td>
<td>&lt;5s</td>
</tr>
<tr>
<td>ZZD</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>39 min</td>
<td>&lt;7s</td>
</tr>
<tr>
<td>XAB</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>&lt;1s</td>
<td>21 min</td>
<td>&lt;3s</td>
</tr>
</tbody>
</table>

The productivity comparison result is shown in Table 4.3. As it can be seen, it takes
around 3 business days (24 hours) to collect the required information for the SZB station
and around 1 business week (40 hours) to model the XAB station. As for stations GZN and
ZZD, the tester failed to generate the valid station model in an acceptable time and aborted
the testing.

On the other hand, the proposed solution can generate station models within 1 hour for
all four stations. Although it cost 15-39 minutes to collect and maintain required configura-
tions, the time consumption is significantly acceptable compared to the conventional manual
approach. Thus, this work can draw the conclusion that the proposed method can bring
notable productivity improvement to the engineering practice.

4.3.4 Applicability Analysis

The ultimate goal of generating a station model is to be utilized in other practical ap-
lications. For example, utilizing this model in Anylogic to build a high-fidelity simulation
Figure 4.8: Anylogic model of XAB model. For another instance, using this model to solve a train platforming and routing optimization problem in Reference [62].

Anylogic is a popular simulation application to build various simulation models in areas of transportation, logistics, manufacturing, and railroad. Its railroad module provides rich APIs for users to generate tracks, connect tracks to switches and build rail networks.

In Anylogic, each track is constructed using a set of shapes - arcs and lines, which allows us to use the graphic information of basic primitives to generate tracks. Secondly, each switch is constructed using a set of tracks if they are connected at endpoints. Finally, a route is defined as a sequenced set of tracks.

Following the above mechanism, the results show that the output can be friendly and accurately read into Anylogic. Figure 4.8 illustrates the result of converting XAB DXF file to a Anylogic railroad station simulation model with the proposed solution.
The proposed solution is also tested to generate models for the train platforming and routing optimization problem. This dissertation refers the reader to [62] all details.
Chapter 5: Conclusions and Future Works

So far, three problems in relation to the railroad station operations have been resolved. The conclusion and future works are as follows.

First, this dissertation proposes a Huffman forest-based anomaly detection (HuffForest) method that tends to overcome three typical scenarios of identifying station outliers according to their performance. The proposed method leverages Huffman encoding to measure the abnormality of the dataset from the perspective of each instance. The measurements are subsequently divided into global measurements and local measurements, which are jointly considered to compute the anomaly score of each instance. Furthermore, a sampling-based version of HuffForest is developed to reduce the time complexity without sacrificing detection accuracy. Experiments are conducted to validate the proposed HuffForest method.

The experiment is conducted on 4 synthetic datasets representing the railroad scenarios and proves that HuffForest can generate robust detection results in scenarios where existing methods are not reliable. The proposed method is also compared to the state-of-the-art iForest and LOF methods on multiple classic benchmarks, where it is demonstrated to outperform the baseline methods on detection accuracy with an acceptable computational complexity.

Although the proposed method shows its advances on 4 synthetic datasets inspired by railroad station anomalies, the conclusion is still not solid enough because of the absence of real-world data. The proposed method is expected to be further validated on realistic data.

Secondly, this dissertation introduces three different interlocking modes that can significantly affect the route conflicts in the TPR and formulated the problem with a simple time-space formulation. Then numerical experiments were conducted to seriously investigate the optimalities of the TPR under each mode and engineering practical suggestions were provided based on the experimental findings.
In summary, both RLSR and SLSR modes notably outperform RLRR mode, where SLSR mode performs stably outstanding for various station and traffic density settings. Significant returns can be obtained when utilizing RLSR mode in large-size stations with high-density traffic volumes. The practical suggestions include: if technology allows, TPR route conflict rules should be modified to fit RLSR mode, and large stations with high-density traffic volumes should be assigned higher priorities. Meanwhile, the development of SLSR should be sustainably supported since its returns in fully utilizing station capacity are really remarkable.

In this dissertation, the TPR problem is solved as a deterministic optimization problem. However, in reality, the stochasticity of train arrivals is common in all scenarios. Thus, it is necessary to investigate the robustness impacts of different modes in the future.

Thirdly, this dissertation proposes an automation framework to convert DXF files to railroad station models. In this framework, a straightforward and practical symbol recognition method is developed dedicated to railroad station graphic symbols. And complete procedures to build undirected railroad station graphs including breaking, clustering, switch recognition, track recognition, graph generation, configuration, and route generation are presented.

The experimental results show that (1) the proposed framework can accurately recognize station symbols and generate graphs and routes as long as the DXF files are constructed under similar engineering drawing rules. (2) the proposed framework can significantly reduce manual work and improve productivity. Minimal user interactions are acceptable with a well-designed GUI application. (3) The output can be easily read and adapted into other advanced applications such as Analogic and application in reference [62].

In terms of future works, the user interaction is expected to be further reduced by investigating techniques to recognize configuration information in DXF files and overcome challenges due to complex railroad symbol variants.
References


Appendix A: Copyright Permissions

The permission below is for the reproduction of material in Chapter 1 Section 1.1 and Chapter 2.

The permission below is for the reproduction of material in Chapter 1 Section 1.2 and Chapter 3.
Author rights

The below table explains the rights that authors have when they publish with Elsevier, for authors who choose to publish either open access or subscription. These apply to the corresponding author and all co-authors.

<table>
<thead>
<tr>
<th>Author rights in Elsevier’s proprietary journals</th>
<th>Published open access</th>
<th>Published subscription</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retain patent and trademark rights</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Retain the rights to use their research data freely without any restriction</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Receive proper attribution and credit for their published work</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Re-use their own material in new works without permission or payment (with full acknowledgement of the original article):</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1. Extend an article to book length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Include an article in a subsequent compilation of their own work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Re-use portions, excerpts, and their own figures or tables in other works.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use and share their works for scholarly purposes (with full acknowledgement of the original article):</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1. In their own classroom teaching. Electronic and physical distribution of copies is permitted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. If an author is speaking at a conference, they can present the article and distribute copies to the attendees</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Distribute the article, including by email, to their students and to research colleagues who they know for their personal use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Share and publicize the article via Share Links, which offers 50 days’ free access for anyone, without signup or registration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Include in a thesis or dissertation (provided this is not published commercially.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Share copies of their article privately as part of an invitation-only work group on commercial sites with which the publisher has a hosting agreement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publicly share the preprint on any website or repository at any time.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Publicly share the accepted manuscript on non-commercial sites</td>
<td>✓</td>
<td>✓ using a CC BY-NC-ND license and usually only after an embargo period.(see Sharing Policy for more information)</td>
</tr>
<tr>
<td>Publicly share the final published article</td>
<td>✓ in line with the author’s choice of end user license</td>
<td>x</td>
</tr>
<tr>
<td>Retain copyright</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>
Optimality analysis of train platforming and routing with different interlocking modes

Author: Yuan Wang, Xiaopeng Li
Publication: Communications in Transportation Research
Publisher: Elsevier
Date: December 2022

© 2022 The Author(s). Published by Elsevier Ltd on behalf of Tsinghua University Press.

Journal Author Rights
Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights