DATA-DRIVEN OPERATIONAL TOOLS FOR FREIGHT RAIL SYSTEMS

By

William Walker Barbour

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

CIVIL ENGINEERING

May 8, 2020

Nashville, Tennessee

Approved:

Daniel B. Work, Ph.D., Chair

Mark D. Abkowitz, Ph.D.

Christopher L. Barkan, Ph.D.

Gautam Biswas, Ph.D.

Craig E. Philip, Ph.D.

ACKNOWLEDGMENTS

I would like to thank numerous collaborators for continuing to provide advice and guidance on various parts of this work.

Prof. Chris Barkan and Dr. Tyler Dick at the University of Illinois at Urbana-Champaign are incredible sources of railroad expertise and were always available to answer any domain questions I might have. Learning from them has been a hugely edifying and enjoyable experience. It is possible that this work, which is very dependent on domain-specific knowledge, would not be possible without them and others at RailTEC.

During my Ph.D., I had the opportunity to study at both the University of Illinois at Urbana-Champaign and Vanderbilt University. Moving between universities was a challenging, but rewarding, experience. I am eternally grateful to everyone at both universities who helped us in the transition and everyone at Vanderbilt who enthusiastically welcomed us into the University. Prof. Mark Abkowitz, Prof. Gautam Biswas, and Prof. Craig Philip have all been great committee members and their feedback was valuable for pushing the boundaries of this work. I am grateful for their service on the committee as well as their continued contribution to my professional development. Prof. Abhishek Dubey and Chinmaya Samal also contributed their time and expertise to the work.

Dr. Shankara Kuppa and others at CSX Transportation have been enormously helpful research partners. They provided their time, expertise, and large amounts of data – all of which are integral to the success of this work.

I would also like to thank my advisor, Prof. Dan Work. He has been a supportive and enthusiastic advisor, more so than I could have ever hoped for. His diversity of knowledge and interest in a broad range of subjects has made my Ph.D. experience ever more engaging. I have explored ideas and areas of study that I did not foresee; I have been challenged at every turn to think technically, but also strategically; and I have grown as a researcher and engineer the whole time. When I graduated from the University of Tennessee, Knoxville, with a degree that emphasized multidisciplinary engineering, not everyone saw that as a strength. Dan realized the benefit of this type of education and helped me apply it in new and inventive ways.

Finally, I would like to acknowledge and thank every one of the undergraduate students and every one of the graduate students I have had the pleasure to work with over the years. Unfortunately, there are just too many to name individually, which just goes to show how many people make a large effort like a Ph.D. possible. But to all of them: you know who you are, you have my eternal gratitude, and thank you!

My time as a Ph.D. student has delivered exactly what I wanted and much, much more. So I am very grateful to all of these many people who made it possible.

I would also like to acknowledge support by the Roadway Safety Institute, the University Transportation

Center for USDOT Region 5, which Includes Minnesota, Illinois, Indiana, Michigan, Ohio, and Wisconsin. Financial support was provided by the United States Department of Transportation's Office of the Assistant Secretary for Research and Technology (OST-R) and by the Federal Highway Administration's Office of Innovative Program Delivery (OIPD), via the Dwight David Eisenhower Transportation Fellowship Program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

I.1 Motivation

Railroads are a major source of freight movement in the United States, particularly for long-distance transportation. The U.S. freight railroads combine to move more freight than any other system in the world (Federal Railroad Administration, 2012). Railroads have become increasingly digitized, from which they derive direct operational benefits, as well as an abundance of new data. These data sources can be leveraged to identify efficiency improvements, perform proactive maintenance, forecast the evolution of the railroad, and much more (Ghofrani et al., 2018) — all of which drives benefits in cost savings, sustainability, and safety.

Large portions of the United States rail network are single-track infrastructure, which introduces operational challenges that are less acute than in double- or triple-track areas. Especially with a heterogeneous mix of trains, which is common in most of the network, physical dynamics can be an additional challenge. Regardless of the track topology in each territory, railroads are continually striving to increase overall train velocity and increase network capacity. Schedule flexibility is an important aspect in many areas of North American railroads, though this is not universal and, in fact, operational philosophies differ on the subject. However, in the presence of any schedule flexibility, it can be more difficult to understand operational performance, train delays, and means by which to improve operations.

Constrained capacity and schedule flexibility combine in many areas to make operations quite unpredictable. Train delays in certain areas can reach multiple hours, which creates management challenges in all aspects of the railroad, as well as for freight customers and passenger trains that share the same infrastructure. This variability and unpredictability is an area of continual improvement, but the fact remains that delays and disruptions cause inevitable, unforeseen issues.

The freight railroads of the future will look, externally, very similar to today and recent past. But internally, data can drive additional decisions, automated systems can take on more responsibility, and these improvements can make for a more efficient rail transportation system.

I.2 Contributions

The main contribution of this dissertation is to provide new data-driven methodologies to support the increasingly data-driven decision-making and operations of North American freight railroads. In a system with inherent variability, difficult physical constraints, and humans in the loop, it is critical to understand how operational decisions affect railroad performance, how they can be improved, and how outcomes can be pre-

dicted. I present here a body of work that helps to address all of these points with an emphasis on providing relevant and useful methods to a complex transportation domain.

The contributions of this dissertation support the overall pipeline of data-driven methods: data preparation, data analysis, and prediction. Clean, workable data is a prerequisite of all data problems from the simplest statistics to the most complex modeling. Analysis of historical data is a powerful process to evaluate systems and improve them. This often involves a combination of computational techniques and human inspection, especially in complex physical systems. Ultimately, prediction or forecasting problems can be an end result of analysis that can improve efficiency and autonomy.

We note to the reader that the specific datasets used to test the methods in this work were collected prior to major operational changes on this particular railroad. However, they still serve as a useful case study to assess the potential of the methods. A trend towards increasing rigidity of railroad schedules via "precision scheduled railroading" techniques may decrease operational variability, but freight traffic continues to be heterogeneous and capacity constrained based on current infrastructure. Railroad operations are, overall, increasingly data-driven and require methods to clean and work with this data, introspection on dispatching practices, and perform real-time and long-horizon prediction.

The specific contributions are as follows:

- **Develop a method called *data reconciliation* that automatically cleans rail dispatch data and imputes missing points. Small errors in data that produce infeasibilities are detected and fixed with the smallest possible deviation to attain a complete and feasible dataset.**
  - Historical, complete dispatch data is synthetically decimated, then imputed and fixed with the data reconciliation method. The fixes are then compared to the actual historical values.
  - Various objective functions for the data reconciliation problem are explored and tested on the synthetically-decimated dataset.
- **Formulate a method that analyzes discrepancies between historical/empirical rail dispatch data and a hypothetical but optimal plan for the same scenario. The method is referred to as the *dispatch analysis problem*.** The method is demonstrated on three distinct analysis problems:
  - Computing lower bound replanned dispatch after the evolution of some empirical dispatching decisions. This reveals a temporal link between the empirical decisions made during dispatching and the added lower bound runtime that is induced, even after optimal replanning.
  - In the presence of an increased runtime lower bound under replanning from empirical decisions, alterations to empirical dispatches are revealed that would reduce the lower bound towards its

globally-optimal value. The impact of small changes are of particular interest as a means to identify potential areas of dispatch improvement or train performance improvement.

  - Isolate the effects of individual train performance on the overall dispatch. Link the effects of a train's own performance to the secondary effects it has on other trains. Those which exhibit outsize secondary effects indicate criticality with respect to the overall dispatch plan.

- **Develop mechanisms for machine learning-based prediction of train estimated times of arrival (ETA).**

  - Independent machine learning models constructed for different prediction spans on the network reveal a changing dependency on individual data features based on network position.
  - Multiple machine learning methods are applied to the ETA prediction problem and highlight random forest regression as an effective predictor which can reduce average prediction error by over 40% compared to baseline methods.

These contributions fit together to support data-driven railroad operations in a manner summarized in Figure I.1. Real-time data streams are gathered constantly from various systems on the railroad, such as train movements, yard records, dispatching, maintenance events, and even weather conditions. Many of these data streams are recorded in historical archives and some are monitored and used in real-time operation systems, such as locomotive assignment, crew assignment, and train scheduling. The contribution of data reconciliation is intended to be fed from raw historical data and serve as the direct data source for other systems desiring clean and complete dispatch data. The work on ETA prediction is an example of a real-time operation system which takes into account real-time data as well as cleaned historical data from data reconciliation. Dispatch analysis methods developed in this work also use the cleaned and complete data source provided by reconciliation and are, along with ETA prediction, a source of information for various consumers on the railroad.

I.3 Organization

The remainder of this dissertation is organized as follows. A literature review spanning works relevant to each successive chapter are surveyed in Chapter II. The data reconciliation problem is presented and results are discussed in Chapter III. The dispatch analysis problem and its applications are formulated, along with case study and results for each application of the method, in Chapter IV. Machine learning ETA prediction methods and results are discussed in Chapter V. The dissertation is concluded in Chapter VI, with discussion of potential future work. Appendix A presents a special case of ETA prediction as it applies to grade crossing safety.

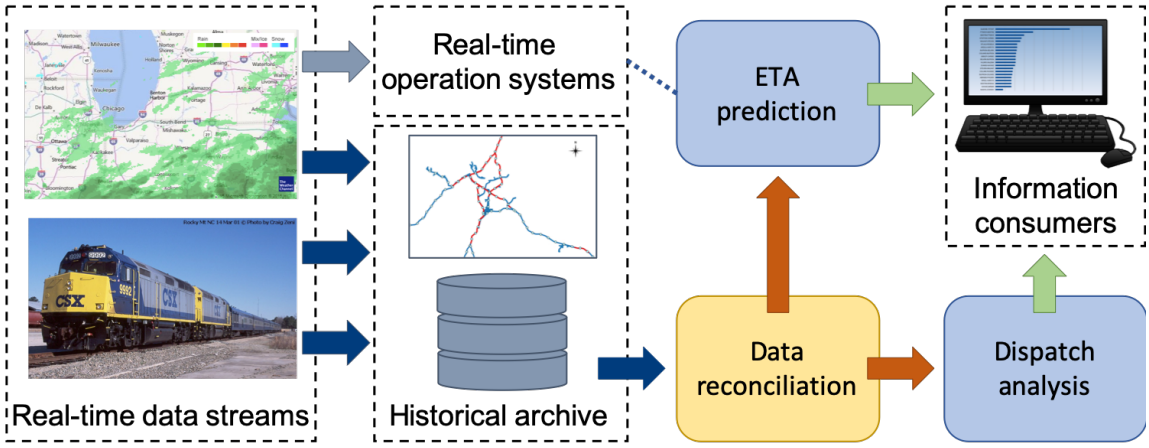


Figure I.1: Summary workflow for data-driven railroad operations covered in this dissertation.

CHAPTER II

Literature review

The development and application of computational technologies on railroads is extensive and spans many aspects of operations, including rolling stock allocation, crew scheduling, maintenance, and, importantly, dispatching. In related work for this dissertation, I first discuss optimization-based rail dispatching, which is used broadly to include long-range planning, periodic scheduling, and real-time control. Second, I summarize works related to the problem of data reconciliation of rail dispatch data. Works on dispatch analysis are then discussed, specifically focusing on replanning or re-optimizing the real-time dispatch problem, analyzing dispatching practices for improvements, and attributing operational delay to contributing sources. Finally, I discuss the body of work related to estimation of train arrival times.

II.1 Optimization-based rail dispatch

Optimization-based rail dispatch is a common tool in passenger and freight railways around the world. Many rail dispatch and control schemes still require humans in the loop, but actions and plans are often suggested by *computer aided dispatching systems* (CAD) (Petersen et al., 1986). Hellström et al. (2003) experiment with a simulation task presented to professional rail dispatchers and discuss the relationship between how dispatchers solve the task and how computer scheduling or dispatching systems often assess the same or similar situations.

These CAD systems are given the physical and logical constraints of the network, which include network topology, speed limits, signalling, train passing logic, and train physics, alongside railroad operating practices and preferences, which include train schedules, train priority, and delay recovery (Wang and Goverde, 2016; Khoshniyat and Peterson, 2015). Numerous scheduling and dispatching objectives exist, such as energy efficiency, speed profile, capacity, and reliability; a selection are discussed in (Narayanaswami and Rangaraj, 2011). Multi-objective optimization schemes are also used (Ghoseiri et al., 2004). The routing problem considers these many factors and constraints and, along with the size of the rail network, results in a large *mixed integer program* (MIP) (Bollapragada et al., 2018; Higgins et al., 1996). These problems are applicable at multiple levels of operations, the hierarchy typically accepted as strategic, tactical, operational, and real-time planning; Törnquist (2006) and Fang et al. (2015) survey works at these various levels of the hierarchy and highlights techniques that have been experimentally validated and those that have been implemented in practice.

One of the first formal definitions of the CAD problem was by Petersen et al. (1986). Higgins et al. (1996)

focused on a similar CAD model as a decision support tool for single-line railways. Murali et al. (2016) used an *integer programming* (IP) model for tactical planning on the Los Angeles rail network. The intractably large MIP problems created by timetabling over a large geographical area and long time horizon are addressed with an incremental heuristic by Gestrelius et al. (2017). Wang and Goverde (2016) took a detailed approach to trajectory optimization from the energy conservation perspective with consideration of train performance calculations. Robust train timetabling was addressed with variable time headways by Khoshniyat and Peterson (2015). Törnquist and Persson (2007) studied the effects of different optimization objectives using a heuristic technique for disturbance re-scheduling on a mixed passenger/freight traffic corridor in Sweden. Bollapragada et al. (2018) describe a modern optimization-based system that handles train dispatching and other ancillary activities used at Norfolk Southern Railway in the United States.

II.2 Data reconciliation

The data reconciliation problem for rail data is an extension of these exact types of dispatch problem formulations that ensures feasibility of operational data that is collected. The integrity of this data is important for derivative analyses performed on it, which have implications in safety, operations, sustainability, and automation. See Ghofrani et al. (2018) for a review of many of the applications in this field. All of these data-driven methods require large volumes of reasonably clean historical dispatch data due to the unique topology of the rail network and the complexity of operations (Wang and Work, 2015; Barbour et al., 2018b; Oneto et al., 2019; Ghofrani et al., 2018).

The number of works focused on data reconciliation is rather limited. An optimization-based data reconciliation problem was introduced by Tjoa and Biegler (1991), where chemical process measurement and control data was studied with respect to noise reduction and gross error correction. Leibman et al. (1992) develop a new method for data reconciliation using nonlinear programming targeted at dynamic and nonlinear environments. Tong and Crowe (1995) introduced the use of principle component analysis for gross error detection in data reconciliation, as an alternative to some statistical tests. Soderstrom et al. (2001) performed gross error detection and data reconciliation simultaneously by formulating and solving a mixed integer linear program for process flows.

In the transportation field, Zhao et al. (1998) used data reconciliation techniques for processing traffic count data under flow conservation constraints. Claudel and Bayen (2011) perform data reconciliation for highway traffic data, posed as a convex program based on constraints derived from a partial differential equation describing conservation constraints.

II.3 Replanning

Unexpected delays, to say nothing of the many potential sources of delay, are inevitable. The ability to limit the size and impact of these delays maintains performance of the railroad and reduces their associated cost. Delays costs the railroads and their customers money and reduces the competitiveness of shipping (and transporting passengers) by rail compared to other modes (Lovett et al., 2015).

Disruption management literature is well-summarized by Fang et al. (2015) and is generally most relevant at the real-time level (Narayanaswami and Rangaraj, 2011). Accordingly, fast algorithms are required if computer aided dispatching is to be useful. Törnquist (2007) develops a heuristic approach for use in replanning under disturbances. Re-scheduling can occur with various objectives and these are assessed on how well they meet performance measures. Minimizing total final delay tends to impose delay on fewer trains and a short planning horizon was shown to be sufficient for longer-term results. An alternative to heuristic methods is an exact decomposition, which is presented in (Lamorgese and Mannino, 2015).

Large disruptions can easily make the timetable infeasible, even with some small adjustment. Corman and D'ariano (2012) construct alternative graphs to serve as decision support in cases of large disruptions and estimate performance metrics of various alternatives. Gestrelius et al. (2012) use optimization to develop schedules that proactively consider schedule alternatives should trains experience delay and require replanning.

When looking at dispatching as a cognitive task performed by a human, it is apparent that dispatchers rely on being able to monitor a large amount of information on train movements, extending beyond the boundaries of their direct control. Dispatchers continually update anticipation and planning based on train movements so they can act proactively. This is critical, whether the railroad runs to a more rigid or flexible schedule. Systems for dynamic planning are identified as important, both in terms of adjusting to train delays and in predicting them (Roth et al., 2001). Pellegrini et al. (2016) perform experiments quantifying the differences in replanning using optimization algorithms and the actual manner in which professional dispatchers handle disruption scenarios. They find potential gains for implementing optimal replanning over current practice.

II.4 Dispatch analysis

As mentioned earlier, delays are inevitable and costly for railroads. The effects of a delay or disruption extend beyond the direct cost incurred by the event because they can influence other trains. This interdependence is generally referred to as delay/disturbance propagation, secondary delay, or knock-on delay. Ultimately, the manner in which a railroad is operated influences its ability to handle delay events. The ability of a railroad to assess its own operating practices can improve its performance and resilience.

II.4.1 Knock-on delay

Knock-on delay occurs when a schedule deviation in one train has a delaying effect on another train; it is also referred to as secondary delay. Railroads differ in their recording of train arrivals and delays, and also have differing thresholds between what is actually considered a delay (Daamen et al., 2009). Isolation and prediction of knock-on delays is difficult because of multiple factors including primary delay magnitude and location, multiple sources of primary delay, timetable of trains, and infrastructure configuration. Three common methods for determining delays are analytical methods, simulation, and empirical statistics (Milinković et al., 2013).

Daamen et al. (2009) find knock-on delays using continually-updated blocking time graphs to determine train conclicts, where the logical and temporal interactions are based in colored Petri nets. Milinković et al. (2013) construct fuzzy Petri nets to isolate train delays using historical data or railroad expertise, when data is not available. Hwang and Liu (2009) uses microsimulation for modeling interactions between trains and measuring delay. Disturbances are introduced to the simulation model of existing scheduled timetable and effects in terms of track occupancy and arrival at stations are measured. They assume that some amount of schedule recovery is available to trains.

Carey and Kwieciński (1994) model relationship between scheduled train headway and knock-on delay. Lovett et al. (2017) quantify the cascading effects and operational costs of slow orders on the railroad, which introduce disturbances to schedules. Hansen et al. (2010) use delay propagation model (timed event graph) informed by historical data for arrival time prediction. Murali et al. (2010) predict delays on aggregated sections of network using results of simulation by Lu et al. (2004), which allows the assessment of scheduling across large networks.

Mussanov et al. (2017) look at the impacts of schedule flexibility on rail line performance. While not an exact quantification of knock-on delay, they find that introducing rigid scheduling of trains does little to affect overall performance until a large portion of operations are run in this manner. Flexibly scheduled trains are, individually, relatively insensitive to the composition (scheduled versus flexible) of traffic. Additionally, schedule flexibility imposes additional infrastructure requirements to maintain the same level of service compared to more rigidly operated schedules (Dick and Mussanov, 2016).

II.4.2 Robust timetabling

Robust timetables are schedules that are able to absorb disruption event without deviation or are able to recover quickly from disruption (Salido et al., 2008). If railroads are trying to adhere to a schedule, such a schedule must be realistic and reliable. Hallowell and Harker (1998) use an analytical line delay model to adjust existing schedules to improve expected punctuality. The work reflects the relationship between

scheduling strategy and performance in terms of punctuality. Lusby et al. (2018) survey the body of timetable robustness literature and note the potential impact of incorporating robustness into the planning and dispatching process.

Railroad operating philosophies differ in North American freight with respect to schedule flexibility. Schedule flexibility, in terms of allowing trains to depart when needed, naturally introduces sub-optimality compared to a prescribed, optimal meet pass plan. In order to allow railroads to operate in real time with greater schedule flexibility, (Sehitoglu et al., 2018) explore the viability of compensating for this effect with greater allowable operating speed, which allows some delays to be mitigated and maintain certain more beneficial schedules.

II.4.3 Relationship to capacity

Train delay is closely related to railroad capacity; one definition states delay as: "the extra time it takes a train to operate on a route due to conflicts with other traffic" (Dingler et al., 2010). Reduction in delay can be achieved by operational changes or infrastructure projects. Dingler et al. (2010) used Rail Traffic Controller (RTC) to simulate traffic scenarios and found, as one delay effect, that opposing direction traffic causing meet events had a much larger effect on delay compared with same direction traffic causing reduced speeds. Within meet events, time spent stopped was the largest contributor to these delays. Additional sidings and double track reduce meet delays, as do equalized priorities. Gorman (2009) used an econometric model to determine causes of delay and the marginal delay impact of adding trains. Results showed that train interactions – meets, passes, and overtakes – contributed most to delay.

The expansion of capacity is a priority for many railroads, both passenger and freight. Making more efficient use of available infrastructure decreases capital costs of expansions. However, when expansions are necessary, dispatching analysis can reveal the ideal locations and configuration (Shih et al., 2014). Railroad capacity can be determined using optimization by maximizing number of additional trains in a timetable or on a line; Abril et al. (2008) develop a tool that can perform capacity analyses in various optimization schemes and present results of the tool with respect to various traffic characteristics. Their tool also generates results on service reliability and schedule robustness. Capacity utilization is inversely related to schedule reliability and punctuality; Yuan and Hansen (2007) analyze the relationship between capacity utilization and the sensitivity of the schedule to disturbances. They find that schedule buffer time decrease is exponentially related to knock-on delay.

II.5 Estimated time of arrival

Several methodologies to produce ETAs are available, including microscopic simulation (Petersen and Taylor, 1982; Şahin, 1999; Marinov and Viegas, 2011), analytical approaches (Assad, 1980), and data-driven techniques (Bonsra and Harbolovic, 2012; Wang and Work, 2015). Due to the complexity of the freight rail network (which limits the accuracy of analytical abstractions) and the difficulty to capture all delay inducing factors in a simulation based model (e.g., decisions made by human dispatchers, special cases involving priority elevation, unplanned maintenance, and weather), a data-driven approach is proposed in this dissertation (Li et al., 2014). This approach is made possible through access to a large and comprehensive freight rail dataset also described in the dissertation. Well designed data-driven techniques are able to generalize to similar but unseen scenarios to those represented in the training dataset, making them useful for prediction of ETAs during typical operations (Marković et al., 2015). Note however that the methods may not extrapolate well to rare and extreme events such as heavy network disruptions, especially when few or no examples exist in the training data.

Many ETA prediction methods for buses and cars also rely on data-driven algorithms (Altinkaya and Zontul, 2013; Mori et al., 2015) similar to those discussed for freight rail. There are, however, fundamental differences in operations between buses and cars, and the rail freight traffic considered in our work. Bus operations are characterized by frequent stops where delays occur due to the passenger boarding and alighting process (Chien et al., 2002). Buses are also delayed en-route between stations due to traffic signals and other vehicular traffic, which are delay factors for cars as well. Importantly, in the bus system, the vehicular traffic represents an external disturbance to the system. Finally, buses and cars are generally physically homogeneous (e.g., they have similar performance characteristics and consequently similar dynamics) to other buses and cars, respectively. These properties are in contrast to freight rail traffic, where the trains are quite heterogeneous with respect to tonnage, power, length, and priority, all of which affect centralized dispatching decisions and, ultimately, delays.

Several lines of research are related to the problem of freight rail ETA prediction. We briefly summarize the most closely related works, and direct the interested reader to the comprehensive reviews available in the works by Bonsra and Harbolovic (2012) and Gorman (2009). The majority of freight trains operate according to a schedule that is constructed in an offline manner and robust to a degree of random, unplanned disturbances (Mu and Dessouky, 2011; Vromans et al., 2006; Khoshniyat and Peterson, 2017). When extreme disturbances cause the original schedule to deteriorate, online rescheduling measures must be implemented to account for the delay and to maintain robustness to further delay (D'Ariano et al., 2007; Hallowell and Harker, 1998; Khadilkar et al., 2017). Numerous efforts are aimed at understanding and quantifying the

causes of delay that influence scheduling, rescheduling, and predictability (Murali et al., 2010; Chen and Harker, 1990; Dingler et al., 2010). Delay is typically formulated in terms of deviation from a train schedule or historical performance, but it can be extended to arrival time prediction for individual trains (Bonsra and Harbolovic, 2012).

Several works have proposed to empirically produce delay or runtime estimates using historical data for passenger rail networks. Kecman and Goverde (2013) propose an ETA prediction framework for passenger rail arrival time prediction using track occupancy data for conflict evaluation. In Kecman and Goverde (2015), track occupancy variables are used along with schedule and delay data in the real-time prediction of running time and dwell time estimation for passenger rail in the Netherlands. Statistical models including robust linear regression, tree-based non-linear regression, and random forests are each applied to running time and dwell time estimation and an emphasis is placed on the importance of location-specific models. Chapuis (2017) uses artificial neural networks to predict arrival times of frequent passenger trains using historical train and station delays. Compared to the proposed work, the ETAs are evaluated in the Netherlands and France, respectively, on high priority passenger traffic (Furtado, 2013; Pouryousef et al., 2015), which also operates with higher punctuality compared to the freight or passenger traffic in the US (Amtrak, 2016). Marković et al. (2015) use support vector regression on passenger railways in Serbia in order to identify relationships between delay at a station and various internal factors (i.e., related to the train and to the railroad) and external factors. The predictive ability of SVR is compared to that of an artificial neural network and is shown to have better performance and maintain interpretability of the model. Wang and Work (2015) estimate passenger rail delays on the Amtrak passenger rail network in the US using vector regression techniques and only historical runtimes between passenger stations. The regression problems are formulated in both a historical and online perspective, but the feature set for prediction is limited and does not contain any data on the freight traffic, which constitutes the majority of traffic on the shared line of road in the US. Online methods presented for passenger rail, accessible because of the data stream created by station arrivals and departures, have not been fully extended to freight rail. Additionally, the magnitude of delay and ETA error for passenger rail is typically on the order of minutes, while delay and ETA error for non-priority freight rail traffic may exceed multiple hours.

The most closely related estimation works on freight trains are the works of Gorman (2009) and Bonsra and Harbolovic (2012). In Gorman (2009), an econometric analysis of free-running and congestion related factors are used to identify the primary causes of delay. The data is partitioned by geographic area and priority groupings. Congestion related factors, such as meets, passes, and overtakes that occur, are found to have the largest effect on delay. Bonsra and Harbolovic (2012) predict runtimes for individual freight trains in an offline setting using regression. Prediction improvements are attained when estimated at the time

of departure. The regression model used train and network parameters and a historical runtime averaging technique for evaluating model performance.

II.6 Summary

The works discussed in the previous sections of literature review constitute a portion of the large body of computational solutions to practical rail transportation problems. Many of the works take data-driven approaches to their respective problems and show that there is substantial potential to use historical data and train scenarios to learn patterns and relationships. These data-driven approaches, however, must still be informed by underlying information about the rail network and train operations. The use of optimization-based dispatching covers a substantial amount of the underlying dynamics of the rail network and suggests a hybrid data-driven approach: one that uses handles rail fundamentals through an optimization model (or otherwise) and uses data-driven techniques such as machine learning on top of this.

CHAPTER III

Data reconciliation of freight rail dispatch data

III.1 Introduction

III.1.1 Motivation

Data-driven methods for railroad operations require abundant, high-quality sources of data for model building. Machine learning, and deep learning methods in particular, require large datasets for training. These methods will learn trends from input data, so if the data contains errors, then the errors may propagate into the trained model and the resulting analysis.

A common challenge in the emerging data science and data analytics fields is the amount of time spent on data cleaning and data preparation. Common tasks include standardizing and normalizing data, identifying faulty data and discarding or correcting it, and imputing values for missing entries. Especially when (reasonably) clean data from large systems that describe physical processes (e.g., freight rail flows) is needed, ad hoc and manual approaches to data preprocessing can easily be inefficient and can often be intractable.

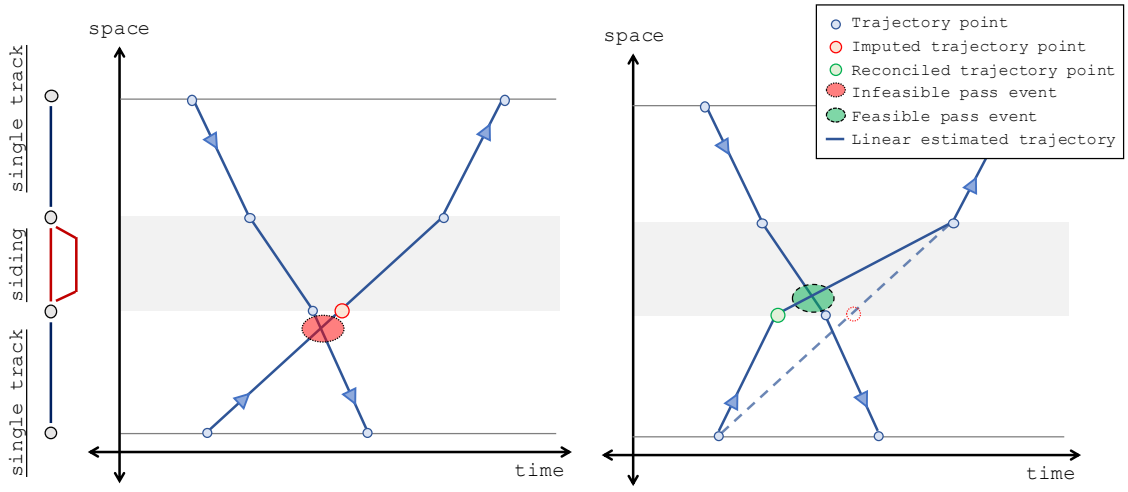
To automate some aspects of data cleaning and data preparation, it is possible to use knowledge about the physical constraints of the system to identify and correct erroneous values. The process by which missing data is estimated and erroneous data is corrected using a model as a constraint is referred to as *data reconciliation* (Tjoa and Biegler, 1991). Given that railroad operations have obvious logical and physical constraints, useful data reconciliation problems can be posed and solved, which we demonstrate in this work.

III.1.2 Overview of data errors in dispatch data

Train trajectory data typically comes in the form of train arrival times at fixed locations on the rail network; in the United States these are often called *on-station points*, or *OS-points*. Most OS-points are located at the endpoints of passing sidings in single-track territory or at crossover points in multi-track territory. Track segments refer to the sections of track delineated by OS-points. The arrival times of a train at each OS-point between two rail yards or terminals on the network constitute a trajectory.

Data errors are identifiable as infeasible trajectories because they violate *meet* constraints (passing events between trains in opposing directions), *overtake* constraints (passing events between trains in the same direction), headway constraints (trains following, meeting, overtaking with insufficient time headway clearance), or other operational constraints. Data may also be missing, e.g., due to incomplete data fusion or sensor failures, which can further compound the difficulty of identifying and correcting errors.

Consider the data error in Figure III.1a, a time-space plot, as an example. Three tracks are shown on the



(a) Infeasible trajectories.

(b) Feasible trajectories.

Figure III.1: A time-space plot example of two trains traveling in opposite directions. Train trajectories are denoted by the blue points and linearly estimated between points. It is obvious that these trains met at the siding track (shaded grey area). This example demonstrates how an erroneous trajectory point (red point) in (a) can result in an infeasible meet location (shaded red oval). This can be corrected by reconciliation of the timing to the green point in (b) to make the meet occur at a feasible location on the siding track (within the grey area).

spatial axis (y-axis): one siding track denoted by the grey shading and two single track segments. Trajectories of two trains traveling in opposing directions are denoted by the blue points, which represent known trajectory points, and the blue lines that are the linearly approximated trajectories between them. In Figure III.1a, there is an imputed point shown in red along the trajectory. The timing value of this point results in a meet event (intersection between trajectories highlighted by the red shaded circle) between the two trains that occurs on a single track segment and is therefore infeasible. It is clear that the two trains must have passed each other on the siding and that the imputed trajectory point must be incorrect. Indeed, by relocating the erroneous imputed point to the location of the green point in Figure III.1b, a feasible set of trajectories is found where the meet event occurs on the siding (highlighted by the green shaded circle). The amount by which the imputed point must be moved is dictated by the safety headway.

This simple example demonstrates one types of data error that can be encountered in rail operations data, noting that in real datasets the errors can be more complex, randomly distributed in the dataset, and can be compounded by missing values. As a consequence, ad hoc or manual approaches to diagnose and fix infeasible data are not scalable to large-scale rail networks that move thousands of trains daily.

III.1.3 Problem statement and contribution

The main contribution of this work is the development of a method to perform optimization-based *data reconciliation* of railroad dispatching data. Given a set of train trajectories and as set of operational constraints, the data reconciliation problem simultaneously corrects any data that is infeasible, and also imputes any missing data. We demonstrate six objective functions and find that the use of $\mathcal{L}_2$ norm and regularization to historical average speeds reduces mean squared error by 60% compared to the original objective.

A constraint set from a dispatch optimization problem that models single-track rail operations is used to perform data reconciliation and we note that the method may be generalized to other optimization-based dispatch tools and network topologies. To illustrate the performance of the method, the data reconciliation problem is implemented on a real freight rail dataset with synthetic omissions in the data.

This is the first work to formalize data reconciliation for cleaning rail dispatch data, which can be a critical step for machine learning and data-driven rail operations and is a practical challenge in the transportation industry.

The remainder of the chapter is organized as follows. Section III.2 provides a general forms of optimization-based dispatching and its relationship to the data reconciliation problem. Section III.3 instantiates a specific data reconciliation problem used in the work on a real dataset from a US Class-1 railroad. In Section III.4, we present and discuss results from applying the reconciliation to the historical dataset and to a synthetically incomplete dataset.

III.2 Background

In this section we first explain the generalized problem formulation of optimization-based dispatching and the corresponding data reconciliation problem formulation.

III.2.1 Optimal dispatch problem

The optimal dispatch problem takes a set of trains traveling on a section of the network (e.g., between major yards or terminals) and finds feasible trajectories that are optimal with respect to minimization of a function of weighted train runtime and satisfy physical and operational constraints. Broadly, many dispatch problems can be posed in the general form:

$$\begin{aligned} \underset{x,z}{\text{minimize:}} \quad & f(x,z) \\ \text{subject to:} \quad & A_1 x + A_2 z \leq b, \end{aligned} \tag{III.1}$$

where the decision variables are $x \in \mathbb{R}_+^p$ and $z \in \mathbb{Z}^q$. In a common formulation, the decision variables x encode times at which trains reach various points on the network, while the integer decision variables z

encode dispatching logic that indicates if and where meets and overtakes occur on the network. The objective function $f(\cdot,\cdot)$ is a performance measure that quantifies the desirability of the dispatch solution, for instance with respect to delay or priority weighted delay of trains. Integer variables may factor into the objective function if, for example, one wishes to minimize the total number of meets and overtakes. The physical and operational constraints, such as the permissible locations of meet and overtake events, headway constraints, and train travel times, are encoded in the inequality constraints $A_1x + A_2z \leq b$. For simplicity the constraints are assumed to be mixed integer linear, although more general dispatch problems can also be considered.

III.2.2 Data reconciliation problem

With a generic form of the optimal dispatch problem defined, it is now possible to define the corresponding data reconciliation problem. The constraint set from the train dispatch problem plays a critical role in the data reconciliation problem. Accurate data reconciliation assumes that the constraint set correctly describes the operations of the rail network. Consider a historical trajectory dataset denoted by $\tilde{x}$ and $\tilde{z}$, possibly containing missing entries. Let $\tilde{x}_\Omega$ and $\tilde{z}_\Omega$ denote the subset of the historical dataset for which entries are present. The data reconciliation problem is written as:

$$\begin{aligned} \underset{x,z}{\text{minimize:}} \quad & g(x_\Omega - \tilde{x}_\Omega, z_\Omega - \tilde{z}_\Omega) + h(x_\Psi, z_\Psi) \\ \text{subject to:} \quad & A_1x + A_2z \leq b, \end{aligned} \tag{III.2}$$

where $x \in \mathbb{R}^p_+, z \in \mathbb{Z}^q$. The variables x_Ω and z_Ω are the subset of the decision variables that correspond to the historical dataset for which entries are present and x_Ψ and z_Ψ are the subset of the decision variables that correspond to missing historical entries.The reconciliation problem finds feasible trajectories, x, z, that are feasible and minimally-perturbed from the historical data according to the performance measure $g(\cdot,\cdot)$. An additional term $h(\cdot,\cdot)$ can be added to the reconciliation problem to further regularize the missing data that must be imputed by the data reconciliation problem. Importantly, while the historical data $\tilde{x}, \tilde{z}$ may or may not be feasible, and may or may not contain missing entries, the reconciled data indicated by the decision variables at optimality, x^*, z^*, are feasible and complete provided the constraint set is not empty.

A variety of possible performance measures can be designed for the data reconciliation problem. For example, a natural choice is an $\mathcal{L}_1$ penalty on the historical data:

$$g(x_\Omega - \tilde{x}_\Omega, z_\Omega - \tilde{z}_\Omega) = ||x_\Omega - \tilde{x}_\Omega||_1, \tag{III.3}$$

which promotes sparsity in the changes to the timing variables from the values in the historical data. In (III.3), we do not consider a penalty on the integer variables z_Ω even though it is possible, because it requires more

care to design and depends on the interpretation of the variables. For example, in the problems instantiated later in this work, the integer decision variables are uniquely determined once the continuous variables are fixed, and the primary objective is to match the timing data as much as possible.

In cases of missing historical data, the design of the regularization term influences the quality of the imputed values found when solving the data reconciliation problem. Supposing again that x denotes timing data, and x_Ψ denotes the vector of entries of x corresponding to the missing data in $\tilde{x}$, one can advance trains as quickly as possible with:

$$h(x_\Psi, z_\Psi) = ||x_\Psi||_1. \tag{III.4}$$

Letting w encode the priority of trains at the various timing points, one can advance the trains based on priority weights:

$$h(x_\Psi, z_\Psi) = w^T x_\Psi. \tag{III.5}$$

It is also possible to regularize based on desired timings x^{des} that allow for encoding desired segment speeds (e.g., average speeds) through the sections with missing data. This can be written as:

$$h(x_\Psi, z_\Psi) = ||x_\Psi - x^{\text{des}}||_1. \tag{III.6}$$

It is also possible to regularize based on the integer variables z_Ψ, to indicate a preference to avoid meets and overtakes, for example.

III.3 Instantiation of a data reconciliation problem

This section provides an overview of the data reconciliation problem formulation including the parameters, decision variables, the objective function, and constraints.

We limit the discussion to terminology and constraints needed to understand the core functionality of the model. For clarity and brevity, in this abbreviated formulation we do not describe end of train clearance timing, trains entering and exiting in the middle of the network section, multi-track segments with crossing tracks, simultaneous meet and overtake events at sidings, and some features unique to this particular network section.

The dispatch optimization and data reconciliation problems share the same parameters, decision variables, and constraint set for a given network topology. Here a specific form of the MIP is used that is based primarily on the dispatch formulation of Petersen et al. (1986) and Higgins et al. (1996), but in principle the data reconciliation problem can be posed using constraints from other optimization-based dispatching problems.

III.3.1 Assumptions

We briefly summarize key assumptions that are made in the data reconciliation problem and comment on their importance.

In this work, the optimization-based dispatching problem is given only at the track segment fidelity. Higher-fidelity data is in use on railroads and also needs to be clean and complete. We do not consider higher-fidelity data here, but note that it is possible to extend these methods without loss of generality. Therefore, this assumption is core to the optimization model presented herein, but simplifying in that the methodology can be generalized to other data streams by relaxing this assumption.

We also assume that train trajectories between OS-points are feasible for a given train, not accounting for train performance beyond the minimum segment travel times discussed later. Given the data fidelity used here, feasibility can only be evaluated on an aggregated level. This assumption is related to the fidelity of data discussed above and must be made for the data used in this formulation.

Lastly, we assume that trains do not make complex maneuvers that include reversing on tracks, splitting or shuffling blocks of rail cars, or using one-way storage tracks for passing. This is a simplifying assumption made to reduce model complexity, and because these maneuvers are generally rare on railroad mainlines.

III.3.2 Problem setup

A track graph for the network section over which trains operate is delineated by OS-points that are located at the endpoints of multi-track segments or siding tracks (as discussed in Section III.1.2). The set of all tracks segments is denoted M, with individual segments assigned integer labels beginning with track zero such that $M : 0, 1, 2, 3, ...$ Track segments containing a siding track or multiple tracks are included in the set S, where $S \subset M$. Each track segment $m \in M$ has length K_m. Trains travel in two directions on the network: direction 1 and direction 2. Define direction 1 to be the direction of increasing track integer labels and direction 2 to be the decreasing direction. Because track segments are denoted by integers, we can refer to the successor track in direction 1 relative to a segment $m \in M$ as segment $m + 1 \in M$. Likewise, the successor track in direction 2 relative to segment m is $m - 1$.

The set of trains traveling in direction 1 is denoted I and direction 2 trains are J. Individual trains are referred to as $i \in I$ or $j \in J$ and have unique identifiers such that $I \cap J = \emptyset$. Each train has a known length denoted L_i (or L_j).

An example network section is shown in Figure III.2. This section has five track segments, $M : \{0, 1, 2, 3, 4\}$, two of which contain siding tracks, $S : \{1, 3\}$. The length of each track segment is labeled K_0, K_1, etc. Two trains are also shown: train $i \in I$ in direction 1 and train $j \in J$ in direction 2.

Additional parameters used in the objective function and in constraints must be provided. Historical data,

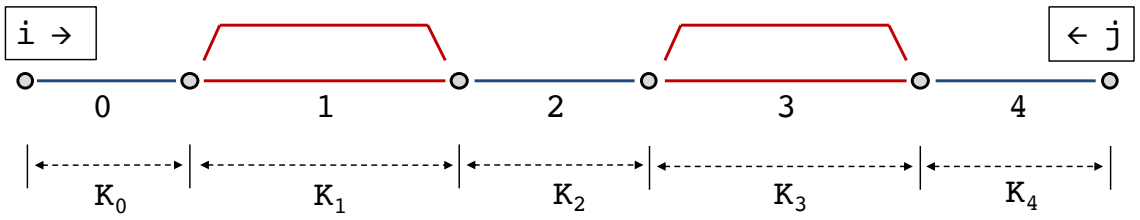


Figure III.2: Depiction of notation used in data reconciliation problem for an example track graph with 5 segments. The set of all track segments in this example is $M : \{0, 1, 2, 3, 4\}$ and the set of siding segments is $S : \{1, 3\}$. The length of each track segment is denoted K_0, K_1, etc. The two trains in this example are labeled $i \in I$, which travels in direction 1, and $j \in J$, which travels in direction 2.

as discussed in Section III.2.2, is denoted $\tilde{x}$. Specifically, we define the historical completion time of each train i (and j) for each track segment m to be $\tilde{x}_{i,m}$ (and $\tilde{x}_{j,m}$). Note that completion time of a segment is relative to direction, so the values $\tilde{x}_{i,m}$ and $\tilde{x}_{j,m}$ for the same segment m refer to different endpoints of the track segment.

The free run (i.e., minimum) traversal time of each track segment is defined specific to each train. If train i takes the main line track on a segment m, its free run traversal time of the segment is $T_{i,m}$. If train i takes the siding track on a segment $s \in S$, its free run traversal time across the siding track is $U_{i,s}$. We assume that the siding free run time values are greater than the corresponding main line free run time (i.e., $U_{i,s} \geq T_{i,s}$). Trains $j \in J$ have corresponding parameters $T_{j,m}$ and $U_{j,s}$.

For meet or overtake events between pairs of trains, we define minimum clearance headways in terms of time (minutes). The minimum headway between trains traveling in the same direction is H_{i_1,i_2} (or H_{j_1,j_2}) for pairs of trains in $i_1, i_2 \in I$ (or $j_1, j_2 \in J$). For trains traveling in opposite directions, the headway time is $H_{i,j}$, where $i \in I$ and $j \in J$.

III.3.3 Decision variables

The real-valued decision variables for the reconciliation problem are the reconciled trajectory timing values. The decision variables representing the reconciled data are denoted $x_{i,m}$ and $x_{j,m}$ for trains $i \in I$ and $j \in J$, respectively, corresponding to each track segment $m \in M$. These correspond to the historical data $\tilde{x}_{i,m}$ and $\tilde{x}_{j,m}$.

The integer-valued decision variables govern the interactions between trains. We use variables indicating train ordering (i.e., the order in which trains complete a track segment) to identify meet and overtake events. Let the set of track segments that are only single-track segments be denoted $M \setminus S$, which is the set M minus the set S. We define the ordering variables $\pi_{i,j,m}$ for all combinations of trains $i \in I$, trains $j \in J$, and track segments $m \in (M \setminus S)$, to be $\pi_{i,j,m} = 1$ if train i crosses segment m before train j, and $\pi_{i,j,m} = 0$

otherwise. For trains traveling in the same direction, we define $\phi_{i_1,i_2,m} = 1$ to indicate that train $i_1 \in I$ completed traversal of segment m before train $i_2 \in I$ (where $i_1 \neq i_2$), and $\phi_{i_1,i_2,m} = 0$ otherwise. Likewise, $\phi_{j_1,j_2,m} = 1$ if train $j_1 \in J$ completed traversal of m before train $j_2 \in J$, where $j_1 \neq j_2$.

The occurrences of meet events are indicated by binary values of $\mu_{i,j,s}$, which take the value $\mu_{i,j,s} = 1$ if a meet occurs between trains $i \in I$ and $j \in J$ along track segment $s \in S$, and $\mu_{i,j,s} = 0$ otherwise. The occurrence of overtake events for trains I in direction 1 are indicated by binary values of $\rho_{i_1,i_2,s}$, which takes the value $\rho_{i_1,i_2,s} = 1$ if a meet occurs between trains $i_1 \in I$ and $i_2 \in I$ (where $i_1 \neq i_2$) along track segment $s \in S$, and $\rho_{i_1,i_2,s} = 0$ otherwise. Values of $\rho_{j_1,j_2,s}$ encode overtakes for trains $j_1, j_2 \in J$ in direction 2.

When meet and overtake events occur, one of the trains in each event must take the siding track and one must take the main line track. Let $\sigma_{i,s} = 1$ if train $i \in I$ used a siding track at track segment $s \in S$, and $\sigma_{i,s} = 0$ if it did not. Likewise, let $\sigma_{j,s} = 1$ if train $j \in J$ used a siding track at $s \in S$, and $\sigma_{j,s} = 0$ if it did not.

III.3.4 Objective function

The specific data reconciliation objective used in the majority of this work is as follows. We apply an $\mathcal{L}_1$ norm on the deviations from the historical data when the historical data is present, and regularize with an $\mathcal{L}_1$ penalty to a background term that encourages trains to travel at a constant speed in all sections for which data is missing. This is written as:

$$||x_\Omega - \tilde{x}_\Omega||_1 + ||x_\Psi - x^{\text{des}}||_1, \tag{III.7}$$

where x_Ω corresponds to a vector containing entries of $x_{i,m}$ and $x_{j,m}$ for all $i \in I$, $j \in J$, and $m \in M$ for which historical data is available. The vector x^{des} is arranged to have entries corresponding to the elements of x for which no historical data is available, and is set assuming trains in the historical dataset travel at constant speeds through sections with missing data, independent of other trains or physical constraints. Note that x^{des} may or may not be feasible, and is only used as a regularization term.

We later, in Section III.4.5, present a comparison of various objective functions with respect to their timing error.

III.3.5 Constraints

III.3.5.1 Travel time constraints

Train timing at each OS-point is governed by prior OS-point timing data and minimum free run times. Precisely, the completion time $x_{i,m}$ for train i of segment m must be greater than or equal to the completion

time $x_{i,m-1}$ of the preceding segment plus the minimum free run travel time $T_{i,m}$ specific to that train and segment. This is written as:

$$x_{i,m} \geq x_{i,m-1} + T_{i,m}, \tag{III.8}$$

where $i \in I$ and $m \in M$. For trains j traveling in direction 2, we have:

$$x_{j,m} \geq x_{j,m+1} + T_{j,m}, \tag{III.9}$$

where $j \in J$ and $m \in M$. Note that the segment preceding segment m is $m+1$ for direction 2, because the track segments labels are numbered in increasing order in direction 1.

When train i uses siding s (i.e., $\sigma_{i,s} = 1$), the completion time $x_{i,s}$ of the track segment s depends on the completion time of the previous segment $x_{i,s-1}$ and the minimum siding travel time $U_{i,s}$:

$$\texttt{IF } \sigma_{i,s} = 1, \texttt{ THEN } x_{i,s} \geq x_{i,s-1} + U_{i,s}, \tag{III.10}$$

where $i \in I$ and $s \in S \subset M$. Recall based on the numbering of the track segments, $s-1 \in M$ refers to the track segment immediately before s and that the siding travel time $U_{i,s} \geq T_{i,s}$, indicating the minimum siding travel time is longer than the minimum main line travel time for each train at each segment.

A similar constraint on the completion time when trains $j \in J$ take the siding track handles trains in the opposite direction:

$$\texttt{IF } \sigma_{j,s} = 1, \texttt{ THEN } x_{j,s} \geq x_{j,s+1} + U_{j,s}. \tag{III.11}$$

III.3.5.2 Meet and overtake constraints

Meet and overtake events are constrained using logical properties of the binary ordering variables π and ϕ.

We constrain the arrival times of opposite direction trains at siding endpoints such that a train may not enter onto a single-track segment until the train in the opposite direction has cleared off the single-track segment, plus a safety headway. Recalling that $\pi_{i,j,m}$ indicates which train (i or j) first traverses a single-track segment $m \in (M \setminus S)$, and takes the value $\pi_{i,j,m} = 1$ if train i traverses first and 0 otherwise. Then the meet constraint is written as:

$$\texttt{IF } \pi_{i,j,m} = 1, \texttt{ THEN } x_{i,m} + H_{i,j} \leq x_{j,m+1}, \texttt{ ELSE } x_{j,m} + H_{i,j} \leq x_{i,m-1}, \tag{III.12}$$

where $m \in (M \setminus S)$, $i \in I$, and $j \in J$. Constraint (III.12) activates based on the value of $\pi_{i,j,m}$ and applies only to single track segments. If $\pi_{i,j,m} = 1$, then train i is arriving at the end of the single track segment before train j, and must have at least $H_{i,j}$ minutes of safety headway before train j proceeds onto the single-track segment. Note that because of directionality, the timing variable $x_{i,m}$ refers to the completion time of the single-track segment by train i and $x_{j,m+1}$ refers to the entry time of train j onto the same single-track segment. In the case that j traverses the single-track segment before train i ($\pi_{i,j,m} = 0$), then we require train j to finish the single-track segment, plus the safety headway, before train i may finish segment $m-1$ and enter onto the single-track segment m. Note that in the case that train j traverses m first, the constraint refers to the opposite end of the single-track segment where the completion time of train j is $x_{j,m}$ and the entry time of train i is $x_{i,m-1}$.

In the case of same-direction trains, we impose a following-headway to the completion times of each track segment depending on which train completes the segment first. Recall that $\phi_{i_1,i_2,m} = 1$ if train $i_1 \in I$ traverses segment $m \in M$ before train $i_2 \in I$, where $i_1 \neq i_2$ (i.e., train i_2 follows train i_1). In this case, the completion time $x_{i_2,m}$ of the segment for train i_2 must be at least H_{i_1,i_2} minutes (the safety headway) after the completion time $x_{i_1,m}$ of train i_1:

$$\texttt{IF } \phi_{i_1,i_2,m} = 1, \texttt{ THEN } x_{i_1,m} + H_{i_1,i_2} \leq x_{i_2,m} \tag{III.13}$$

A similar constraint handles the headway separation of trains traveling in direction 2:

$$\texttt{IF } \phi_{j_1,j_2,m} = 1, \texttt{ THEN } x_{j_1,m} + H_{j_1,j_2} \leq x_{j_2,m} \tag{III.14}$$

The next set of constraints allows overtakes only on siding segments, by forcing the order of same-direction trains to stay the same on single-track segments. For direction 1:

$$\phi_{i_1,i_2,m} = \phi_{i_1,i_2,m-1}, \tag{III.15}$$

where $m \in (M \setminus S)$, and direction 2:

$$\phi_{j_1,j_2,m} = \phi_{j_1,j_2,m+1}, \tag{III.16}$$

where $m \in (M \setminus S)$.

In a single-track network topology with a high volume of traffic, simultaneous meet and overtake events occurring at sidings with more than two parallel tracks do occur, albeit rarely. For example, if a train $i_1 \in I$ is overtaken by train $i_2 \in (I \setminus \{i_1\})$ and both i_1 and i_2 meet train $j \in J$, then three parallel tracks are required. To simplify the presentation, here we only describe the constraints that consider the case of two parallel tracks. Extensions to there or more parallel tracks result in additional meet and pass constraints that are tedious but also result in mixed integer constraints.

Recall that meet events are identified by $\mu_{i,j,s} = 1$ if a meet occurs between trains i and j at siding segment s, and $\mu_{i,j,s} = 0$ otherwise. Overtake events are identified by $\rho_{i_1,i_2,s} = 1$ if an overtake occurred between trains i_1 and i_2, and $\rho_{i_1,i_2,s} = 0$ otherwise. Consider train i_1 at track segment s. The total number of meet events train i_1 experiences with any opposite direction trains in J at s is $\sum_{j \in J} \mu_{i_1,j,s}$. Similarly, the total number of overtakes that train i_1 experiences with any same direction trains $i_2 \in (I \setminus \{i_1\})$ is $\sum_{i_2 \in (I \setminus \{i_1\})} \rho_{i_1,i_2,s}$. To avoid simultaneous meet and/or overtake events occurring on segment s, we would require:

$$\sum_{j \in J} \mu_{i_1,j,s} + \sum_{i_2 \in (I \setminus \{i_1\})} \rho_{i_1,i_2,s} \leq 1 \tag{III.17}$$

where $i_1 \in I$ and $s \in S$.

Likewise, for a train j_1 traveling in direction 2, we have an analogous constraint:

$$\sum_{i \in I} \mu_{i,j_1,s} + \sum_{j_2 \in (J \setminus \{j_1\})} \rho_{j_1,j_2,s} \leq 1 \tag{III.18}$$

with $j_1 \in J$ and $s \in S$.

III.3.5.3 Siding assignment constraints

For each meet event and overtake event that occurs, one of the trains must be assigned to take the siding track, which in turn imposes the minimum siding travel time constraint. These constraints are activated by the values of μ and ρ that indicate the occurrence of meets and overtakes, respectively.

Recall that the siding track indicator variable $\sigma_{i,s}$ takes the value $\sigma_{i,s} = 1$ if train i takes the siding track on segment s, and $\sigma_{i,s} = 0$ otherwise. The same is true for train j and the $\sigma_{j,s}$ variables. When $\mu_{i,j,s} = 1$, a meet occurs between trains i and j at siding segment s. As a result, one and only one of the siding indicator variables $\sigma_{i,s}, \sigma_{j,s}$ must be 1. This is written as:

$$\texttt{IF } \mu_{i,j,s} = 1, \texttt{ THEN } \sigma_{i,s} + \sigma_{j,s} = 1, \tag{III.19}$$

where $i \in I$, $j \in J$ and $s \in S$.

Likewise, for overtakes occurring in direction 1 between trains $i_1, i_2 \in I$ on siding $s \in S$ (indicated by the value $\rho_{i_1,i_2,s} = 1$), we have:

$$\texttt{IF } \rho_{i_1,i_2,s} = 1, \texttt{ THEN } \sigma_{i_1,s} + \sigma_{i_2,s} = 1. \tag{III.20}$$

A similar constraint holds for overtaking trains in direction 2:

$$\texttt{IF } \rho_{j_1,j_2,s} = 1, \texttt{ THEN } \sigma_{j_1,s} + \sigma_{j_2,s} = 1, \tag{III.21}$$

where $j_1, j_2 \in J$ and $s \in S$.

Finally, any trains using siding tracks must be short enough to physically fit on the available track length without interfering with switch points at the end of the siding track. If the length L_i of a train $i \in I$ is greater than the length K_s of a siding segment $s \in S$, then train i must not be assigned to take siding s (i.e., $\sigma_{i,s} = 0$). This is written as:

$$\texttt{IF } L_i > K_s, \texttt{ THEN } \sigma_{i,s} = 0, \tag{III.22}$$

with a similar constraint holding for trains $j \in J$:

$$\texttt{IF } L_j > K_s, \texttt{ THEN } \sigma_{j,s} = 0. \tag{III.23}$$

We note that variables identifying meets μ and overtakes ρ are set by additional constraints using logic derived from timing variables, which we do not enumerate here. Similar sets of constraints are also used to encode the IF/THEN/ELSE logic used to simplify the presentation of the constraints. The complete problem formulation results in a mixed integer optimization problem and does not require the use of a constraint programming solver.

III.4 Data reconciliation case study on US class-1 freight rail data

In this section, the data reconciliation problem from Section III.3 is run on data from a portion of a US class-1 freight railroad network. First, a description of the historical dataset and computational environment on which the data reconciliation problem is implemented are described. Two sets of experiments are run to assess the quality of the data reconciliation approach. In the first experiments, the data reconciliation problem is applied to a dataset that is complete but contains errors, for example due to upstream data cleaning steps to impute missing values. In the second set of experiments a synthetic dataset is created from the real original dataset by decimating entries of the complete dataset. Since the true entries are known, it allows assessment

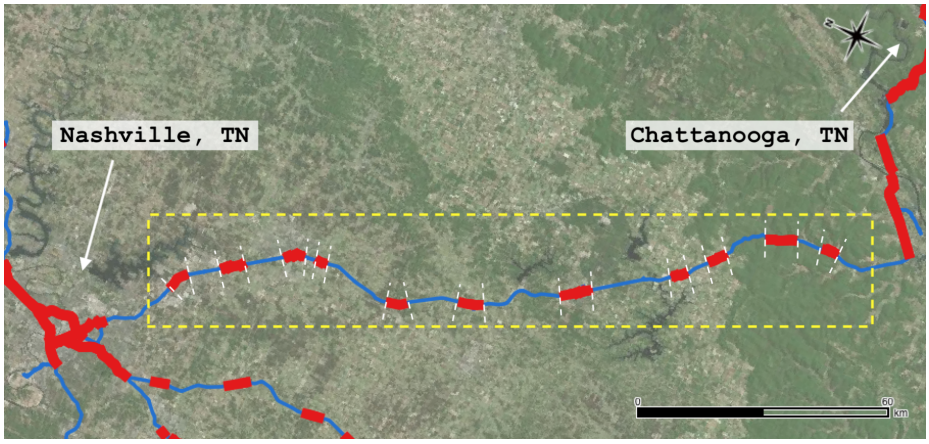


Figure III.3: Map of the rail network territory is shown in the yellow dashed box between Nashville, TN, to Chattanooga, TN, United States. Multi-track sections are shown in red and single-track sections are shown in blue. The scale bar represents 60 kilometers.

of the quality of the imputed solutions from the data reconciliation problem.

III.4.1 Description of historical dataset

The experiments described in this section use a real historical dispatch dataset from a section of the CSX Transportation rail network in the eastern United States between Nashville, TN, and Chattanooga, TN, described also in Barbour et al. (2018a,b). The time period used is six months between January 1, 2016, and June 30, 2016. The dataset contains 4368 hours of data and it includes more than 3,000 individual train trajectories. This section of the network is approximately 100 miles in length (160 km) and is highlighted by the yellow dashed box in the map in Figure III.3. The test corridor is predominantly single track (blue sections on the map) with 11 passing sidings (red sections with dashed line delineations) of varying length. It is a highly congested area of the CSX network and trains must also contend with significant grade at multiple locations caused by mountains. The topology of the network combined with the high volume of traffic result in many meet and overtake events.

III.4.2 Computational environment

The data reconciliation problem is written in the AMPL mathematical programming language and solved using CPLEX 12, a commercial MIP solver. The model is connected to Python code that loads and transforms data, extracts results, and analyzes the output.

In order to maintain a reasonable size of MIP for the reconciliation problem, the data reconciliation problem is solved for datasets in a sliding window with a length between 8 and 24 hours (exact values explained in Section III.4.4). A single 24 hour dataset containing approximately 20 trains yields a MIP of

approximately 5,000 variables and 20,000 constraints, of which approximately 4,000 variables are binary and approximately 15,000 constraints encode logical constraints between the binary variables.

III.4.3 Experiment 1: reconciliation of a complete but erroneous historical dataset

The first set of experiments are conducted on the six-month long historical dataset, which is complete, but contains errors. Any missing data points are imputed in upstream data cleaning steps which may or may not result in feasible trajectories. Using this dataset we apply the data reconciliation problem to identify and automatically correct erroneous data that do not satisfy operational constraints. The complete dataset is analyzed in a 12-hour shifting window until all data has been reconciled.

The results are as follows. On average, each 12-hour window of raw historical data contains approximately three errors that are corrected by the data reconciliation problem. Due to the proprietary and sensitive nature of the historical dataset, detailed descriptions and analysis of the errors (e.g., statistics on the types and the frequency at which they occur) specific to this dataset are not discussed in depth here. To qualitatively assess the quality of the reconciled data, after application of the data reconciliation problem, one week of the historical and reconciled data is manually inspected. The manual inspection verified that the reconciled data only deviates from the historical data in places where the historical data led to constraint violation. Common errors identified during manual inspection include infeasible meets and passes and headway constraint violations due to errors in the timing data.

To give an insight into the type of errors that are automatically corrected, a representative example of an error in the historical data is shown in Figure III.4 (The first eight hours of the 12 hour window are shown). Sidings and multi-track segments, where trains may pass each other, are denoted as grey shaded areas, with the white areas denoting single-track segments. The historical train trajectories (blue lines) have impermissible meet/overtake events that are magnified in the figure inset. The errors are evident in the stringline diagram because the expected meet and overtake events (i.e., the intersection point between trajectories) occur on a single track segment. In contrast, the data reconciliation problem produces the same trajectories as the historical dataset everywhere except in the neighborhood of the infeasible meet/overtake events. In that area, the reconciled data is indicated by the red line, and it results in a set feasible trajectories for all trains. There are three tracks at this passing siding, allowing both a meet and an overtake event to occur simultaneously. Note in Figure III.4 that trajectories that do not cover the entire space correspond to local trains that complete routes between small intermediate destinations on this section of the network.

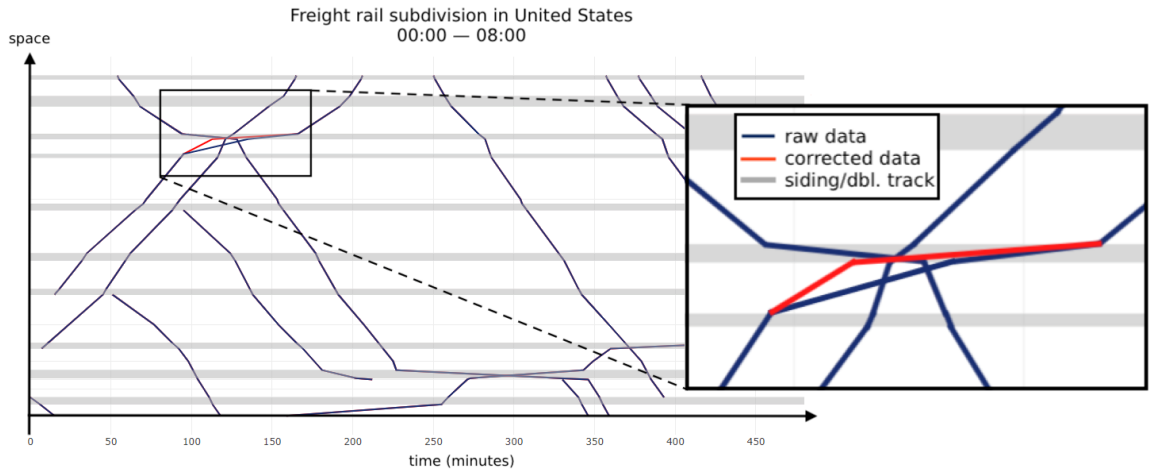


Figure III.4: Stringline diagram of historical and reconciled data. Sidings and multi-track segments are shown as grey shaded areas. Raw train trajectory data is shown as blue lines. The raw data indicates that two meet and overtake events, magnified in the figure inset, occur on a single-track segment, which is infeasible. The red line is the reconciled data that results in feasible trajectories.

III.4.4 Experiment 2: reconciliation of a synthetically decimated historical dataset

Next we quantitatively assess the performance of the data reconciliation problem when imputing missing data with feasible values. We begin with the historical data and create a dataset with missing entries by decimating (removing) a subset of the data entries. This is done to allow comparison between the imputed values produced by the data reconciliation problem with the true historical values that are known (but decimated in the data given to the data reconciliation problem).

To aid in interpretability of the results, the data is decimated only in areas far from any infeasible portions of the historical data, i.e., the historical data that is decimated is feasible. We clarify this is not a limitation of the method (i.e., it can be applied to a dataset containing both missing and erroneous data), but that it is not trivial to assess if differences between the imputed and historical data are due to infeasibility of the historical data, or due to a poor imputed result from the data reconciliation problem. In the experiments conducted next, the synthetically decimated data is feasible so the ambiguity is avoided.

III.4.4.1 Generation of synthetically decimated historical datasets

The synthetically decimated historical dataset is created by removing known trajectory points around meet and overtake events in the reconciled historical data. At each of these events, a particular number of data points (per train) immediately before and immediately after the meet or overtake event are removed. This results in missing data centered around known meet and overtake events. Figure III.5 shows an illustration of this removal process for a meet event between two trains. One point before the meet event in each trajectory

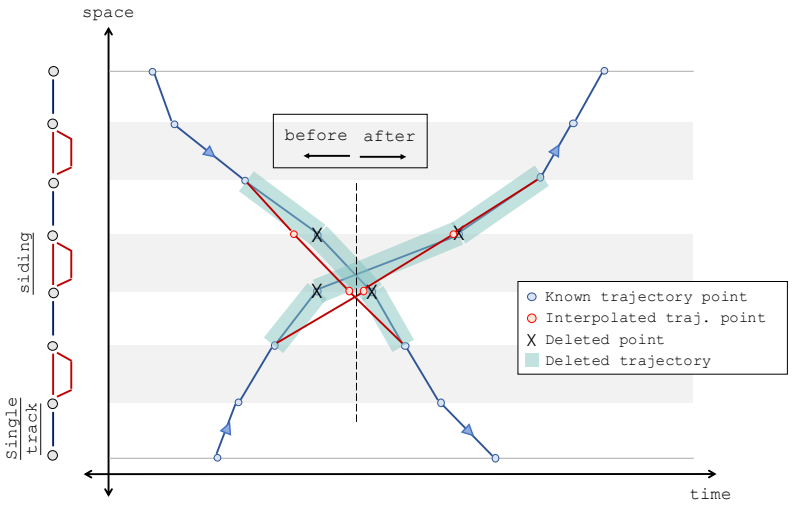


Figure III.5: Four known trajectory points are selectively removed around meet/overtake events that are identified in historical data. One point immediately before the event and one point immediately after the event are removed for each train. These deleted points are shown as black 'X' markers and the missing trajectory segments are highlighted in light blue. A linear interpolation to impute the missing data (red points and lines) can result in infeasible meets.

and one point after the event in each trajectory are removed.

We assess and compare the quality of the corrected and imputed data from data reconciliation with imputed data from a naive linear interpolation approach. In Figure III.5, the red lines and points represent the values imputed via linear interpolation. The interpolation uses the nearest known trajectory points to calculate the average speed across the missing trajectory section, from which the missing points are interpolated. There are many methods more complex than linear interpolation to which data reconciliation could be compared – speed-regularized interpolation, delay minimization, and energy conservation, to name a few – but linear interpolation is used here as a straightforward baseline method.

The quality of the imputed trajectory points is assessed by *i*) evaluating the location at which the recovered trajectories estimate meet and overtake events to occur and *ii*) calculating the time difference between each imputed value and the known trajectory value.

The location of each meet or overtake event found in the reconciled data is *feasible* if and only if it is on a siding or multi-track segment and does not violate other constraints. This location is *correct* if it matches the true location of the event indicated by the known data. Note it is possible for linear interpolation to produce feasible or infeasible, and correct or incorrect imputed values. In contrast, data reconciliation always produces feasible imputed values which may or may not be at the correct location.

The quality of the timing data is assessed via the *mean absolute error* (MAE) and *mean squared error* (MSE) of the imputed values compared to the historical values that are decimated. Letting x^*_Ψ denote the

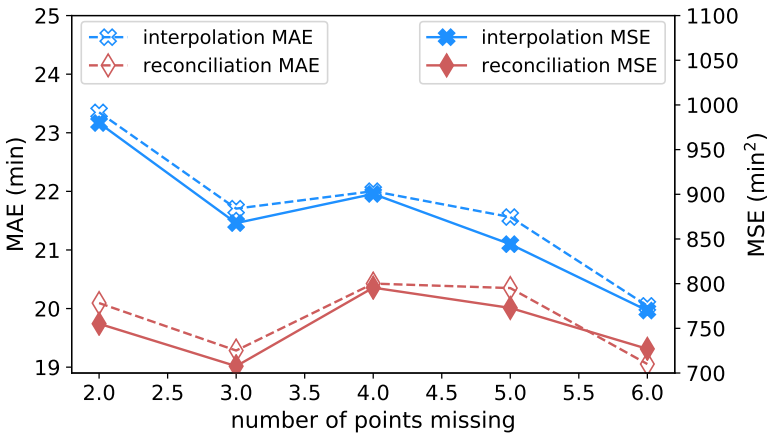


Figure III.6: Mean absolute error and mean squared error of timing values for each missing point imputed by interpolated and reconciliation. MAE and MSE are averaged across trials, grouped by the total number of missing points around each meet or overtake event.

vector of reconciled values, and with a slight abuse of notation, let $\tilde{x}_\Psi$ denote the historical data which is known but synthetically decimated in the experiment (i.e., the assumed ground truth). The quality of the imputed values are:

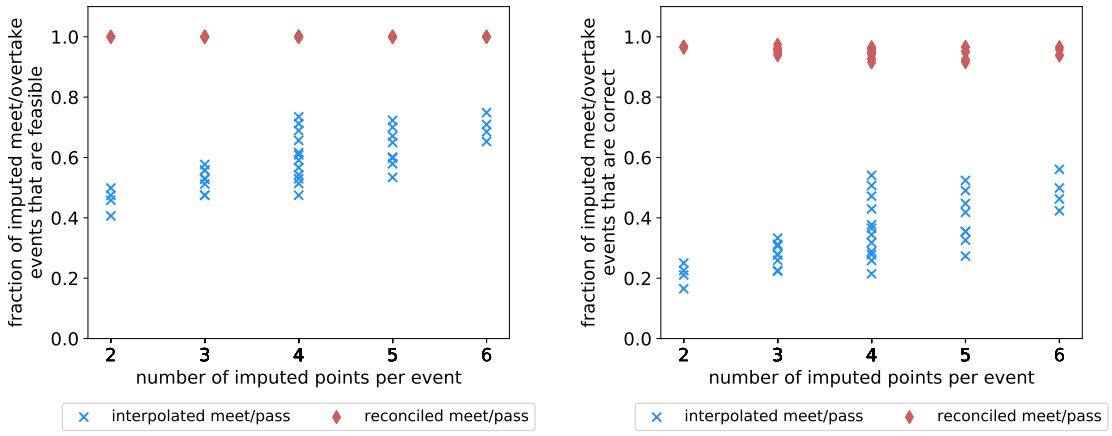
$$\text{MSE} = \frac{1}{|\tilde{x}_\Psi|}||\tilde{x}_\Psi - x^*_\Psi||_2^2, \qquad \text{MAE} = \frac{1}{|\tilde{x}_\Psi|}||\tilde{x}_\Psi - x^*_\Psi||_1, \tag{III.24}$$

where $|\tilde{x}_\Psi|$ denotes the number of imputed values.

III.4.4.2 Results on synthetically decimated datasets

The results of the data reconciliation experiments on the synthetic, incomplete dataset are presented next. A total of 45 data reconciliation experiments are conducted on the six month dataset. Each experiment is defined by *i*) the number of points per train that are removed immediately before a meet or overtake event, *ii*) the number of points per train that are removed immediately after a meet or overtake event, and *iii*) the length of the sliding window. For example, the first experiment removes a single point per train before and a single point per train after each meet/overtake event, and the data reconciliation problem is solved on a sliding eight hour window through the six month dataset. The remaining experiments are defined by considering: *i*) the number of missing points per train immediately before a meet/overtake event (1, 2, or 3 points), *ii*) number of missing points per train after an event (1, 2, or 3 points), and *iii*) the sliding window length (8, 12, 16, 20, or 24 hours).

The MAE and MSE for trajectory points imputed by data reconciliation and linear interpolation are shown in Figure III.6. The results are grouped by the number of total missing points around each meet or overtake



(a) (b)

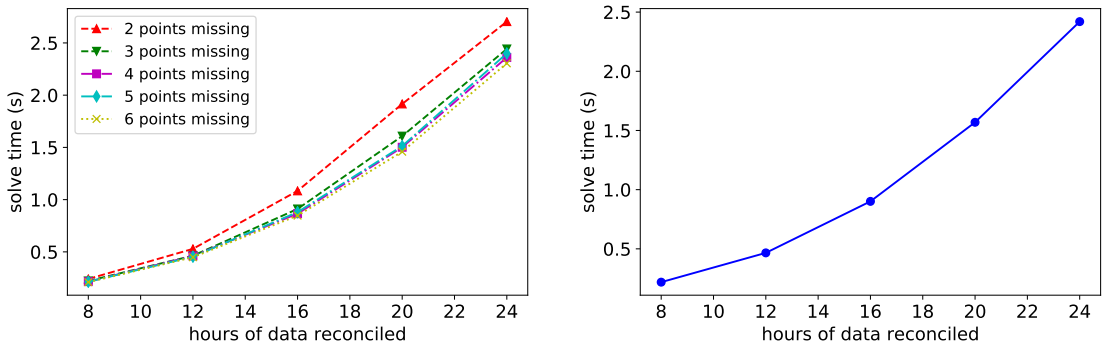
Figure III.7: Fraction of meet/overtake events found by data interpolation and reconciliation that are (a) found to occur at a feasible location, and (b) at the correct location.

event (i.e., the total points immediately before and after each event, resulting in between two to six missing points). Data reconciliation results in a 5-15% reduction in both MAE and MSE compared to linear interpolation.

The fraction of meet and overtake events that are found to occur at a feasible location when imputed by data reconciliation and by linear interpolation are shown in Figure III.7a. The trial results are grouped by total number of points missing (i.e. points immediately before and after an event). The multiple values for a given number of points correspond to the various experiments run with differing points missing before or after each event but resulting in the same number of total missing points per event. Because the data reconciliation problem uses the physical constraints when interpolating the points, 100% of the imputed meet/overtake events are feasible. In contrast, linear interpolation results in feasible meet and overtake locations in only 40-70% of cases and exhibits variability across the different experiments for the same number of total missing points.

Figure III.7b shows the fraction of meet and overtake events that are estimated to occur at the correct location as indicated by the known data, grouped by the number of missing points around each event. Reconciliation recovers the correct location for meet and overtake events in approximately 95% of cases, while linear interpolation recovers only 20-50%. Additionally, reconciliation preforms consistently across trials, with interpolation demonstrating higher variability in performance.

The reconciliation problem executes very quickly, even on large amounts of data. Solve time increases non-linearly as a function of the number of hours used for the shifting window, as seen in Figure III.8b. The number of missing points per overtake event does not have a substantive effect on the solve time, as shown



(a) Average solve time, in seconds, by amount of missing data at each meet and pass event.

(b) Average solve time, in seconds, across all trials and amounts of missing data.

Figure III.8: Solve time of data reconciliation model by length of shifting data window.

in Figure III.8a. The solve times for two and three missing points per meet/overtake event are slightly longer than trials with larger numbers of missing points, but follow a similar trend to the larger numbers of missing points. Solve time of the reconciliation model is low due to the fact that the majority of constraints are already satisfied and the number of corrections required between historical and reconciled data is low. Based on the solve time for the reconciliation model, a year of data from a large rail network (e.g., track networks of freight railroad companies in the United States) could be reconciled in about 20 hours of total CPU time with a 24 hour sliding window.

III.4.5 Experiment 3: comparison of objective functions for reconciliation

In this experiment, we assess a selection of objective functions on the data reconciliation problem with respect to their ability to reconstruct the synthetically decimated dataset described in Section III.4.4 with minimal timing error on trajectory points. The performance of particular objective functions may differ by dataset, but this survey is intended to benchmark performance of some candidate functions.

Each of the six candidate objective functions enumerated in Table III.1 (the first of which is used in the other experiments throughout this work) is tested on a 12-hour shifting window of the same dataset with four trajectory points, two upstream and two downstream of each meet and pass event, synthetically decimated. Total mean absolute error and mean squared error were calculated by comparing the reconciled trajectory points to the trajectory points before decimation in each 12-hour window. MAE and MSE are calculated per trajectory point across the full six months of the dataset.

The results of the objective function comparison in terms of trajectory point MAE and MSE are shown in Figure III.9. Three different methods are used to regularize imputed data according to the variables x^{cs}, x^{ss}, and x^{tt}. The introduction of historical average segment speed (x^{ss}) reduces MAE and MSE by approximately

objective function	equation	definitions
$\mathcal{L}_1$, constant speed	$\|\|x_\Omega - \tilde{x}_\Omega\|\|_1 + \|\|x_\Psi - x^{\text{cs}}\|\|_1$	x^{cs} is interpolated assuming constant speed across track segments with missing data.
$\mathcal{L}_2$, constant speed	$\|\|x_\Omega - \tilde{x}_\Omega\|\|_2 + \|\|x_\Psi - x^{\text{cs}}\|\|_2$	x^{cs} – see previous.
$\mathcal{L}_1$, average segment speed	$\|\|x_\Omega - \tilde{x}_\Omega\|\|_1 + \|\|x_\Psi - x^{\text{ss}}\|\|_1$	x^{ss} is interpolated assuming track segment travel times are distributed proportional to average historical segment speeds.
$\mathcal{L}_2$, average segment speed	$\|\|x_\Omega - \tilde{x}_\Omega\|\|_2 + \|\|x_\Psi - x^{\text{ss}}\|\|_2$	x^{ss} – see previous.
$\mathcal{L}_1$, train type average segment speed	$\|\|x_\Omega - \tilde{x}_\Omega\|\|_1 + \|\|x_\Psi - x^{\text{tt}}\|\|_1$	x^{tt} is interpolated assuming track segment travel times are distributed proportional to average historical segments speeds, grouped by corresponding train type.
$\mathcal{L}_2$, train type average segment speed	$\|\|x_\Omega - \tilde{x}_\Omega\|\|_2 + \|\|x_\Psi - x^{\text{tt}}\|\|_2$	x^{tt} – see previous.

Table III.1: List of objective functions used for comparison, including description of variables used for missing data imputation.

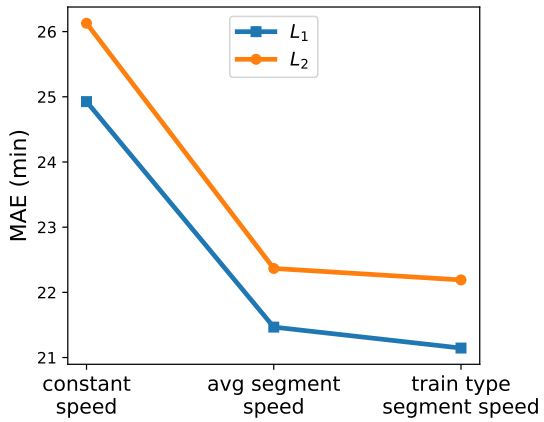
15% for both $\mathcal{L}_1$ and $\mathcal{L}_2$ cases. Given the higher variability of this network section relative to others, the effects of this segment-level speed variable may differ. Refining the average segment speed according to train type does not have a substantive impact. Introducing historical train performance will likely enhance reconciliation in most cases because it better captures the timetable characteristics.

When comparing the $\mathcal{L}_1$ and $\mathcal{L}_2$ cases within each regularization method, we see that the $\mathcal{L}_2$ objective reduces MSE by over 50% (Figure III.9b), at the expense of a slight increase in MAE of 4-5% (Figure III.9a). This is to be expected for the minimization of the $\mathcal{L}_2$ norm, which reduces large deviations for corrected data points by introducing small, additional deviation (compared to the $\mathcal{L}_1$ objective) for points that have small corrections.

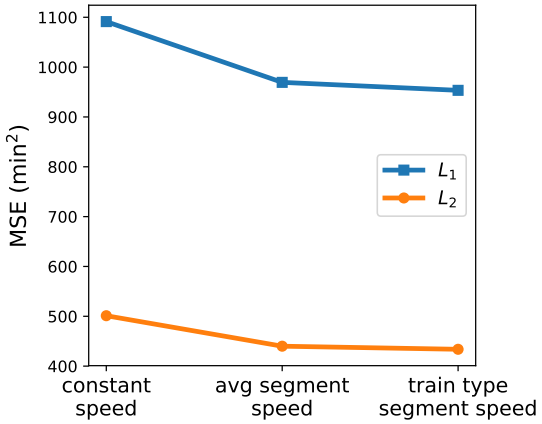
Ultimately, we believe that the large reduction in MSE of $\mathcal{L}_2$ objective functions justifies the small increase in MAE and that the introduction of historical data is a worthwhile addition to the reconciliation methodology. We therefore make the recommendation for future use (at least based on this dataset), of the $\mathcal{L}_2$ objective function with regularization according to the historical average segment speed, x^{ss}.

III.5 Conclusion

Given the growing emphasis on data driven analysis and algorithms to improve operational efficiency, tools are needed to automate the cumbersome data cleaning process. This work introduced the data reconciliation methodology as a tool to correct errors and impute missing values in operational rail datasets. The data reconciliation problem leverages operational constraints that are commonly used in dispatch optimization in a new context that enables efficient reconciliation of infeasible historical data. To demonstrate the viability of



(a) Comparison of mean absolute error (MAE), per trajectory point.



(b) Comparison of mean squared error (MSE), per trajectory point.

Figure III.9: Error results of objective functions across $\mathcal{L}_1$ and $\mathcal{L}_2$ versions of each method for regularizing imputed data points: constant speed (x^{cs}), average historical segment speed (x^{ss}), and average historical segment speed by train type (x^{tt}).

the method, the data reconciliation method is instantiated and applied to a real six-month dataset containing several thousand trains on a complex portion of a US Class-1 rail network. The data reconciliation problem is found to identify and correct erroneous data, as well as impute missing data in a way that is always feasible and often correct.

Numerous extensions to the data reconciliation problem are possible. For example, a detailed design and comparison of different performance measures in the data reconciliation problem objective function might lead to improved accuracy of the reconciled data. It will also be interesting to investigate the sensitivity of the data reconciliation problem to different constraint formulations. In addition to the optimization model discussed in this work, we also intend to test the data reconciliation model on an optimization-based dispatching formulation for multi-track network topologies. Finally, we note that the data reconciliation problem posed here does not identify inefficient but operationally feasible errors. Extensions to identify these errors would be a valuable addition to the rail data cleaning toolbox.

CHAPTER IV

Dispatch analysis

IV.1 Introduction

IV.1.1 Motivation

The challenges of rail network congestion and efficiency motivate the need to answer critical questions about how trains are dispatched and how the railroad state will evolve through time. This analysis is made possible by increasing availability of railroad data. The trends toward better automation and forecasting are continuing, but delays and deviations from the operating plan will persist due to realities that include weather, mechanical failures, and train heterogeneity. There remains the need to improve schedules and dispatching to reduce sources of delay and the variability they cause. Particularly in single-track territories, maintaining a schedule or operating plan is difficult with capacity pressure on the network.

Many prior works have addressed questions about impactful railroad operational practices such as propagation of train delay, impact of disturbances, robust scheduling, and replanning in the presence of delays and disturbances, to name a few (Hansen et al., 2010; Milinković et al., 2013; Lusby et al., 2018; Fang et al., 2015).

Some findings are generalizable across the rail network and various traffic compositions (Mussanov et al., 2017; Sehitoglu et al., 2018; Yuan and Hansen, 2007), but deploying tools to analyze specific dispatching and scheduling practices has the potential to reveal more detailed findings. Certain periods of time, such as particularly problematic dispatching scenario or instances when operations ran better than normal, are useful to analyze in detail. What went right or what went wrong can be revealed and this insight can inform future strategies.

Some dispatch analysis inquiries, generally, can be performed using micro-simulation (Dingler et al., 2009; Dick and Mussanov, 2016; Mussanov et al., 2017). A simulation environment can emulate dispatching and train movements given a network state and schedule, even incorporating random delay and robustness in some cases. This is a powerful tool for what-if analysis and exploring alternative scenarios; examples include the impact of added trains on a corridor, siding and infrastructure placement, and the impact of train characteristics. Rail Traffic Controller is one such simulation environment that incorporates this high level of fidelity; one of its key features is its conflict resolution logic, used for dispatching trains (Tobias et al., 2010).

Some of the limitations or challenges with simulation-based analysis are optimality conditions, scenario exploration, and cost quantification. Most simulation environments do not run a routine that guarantees global

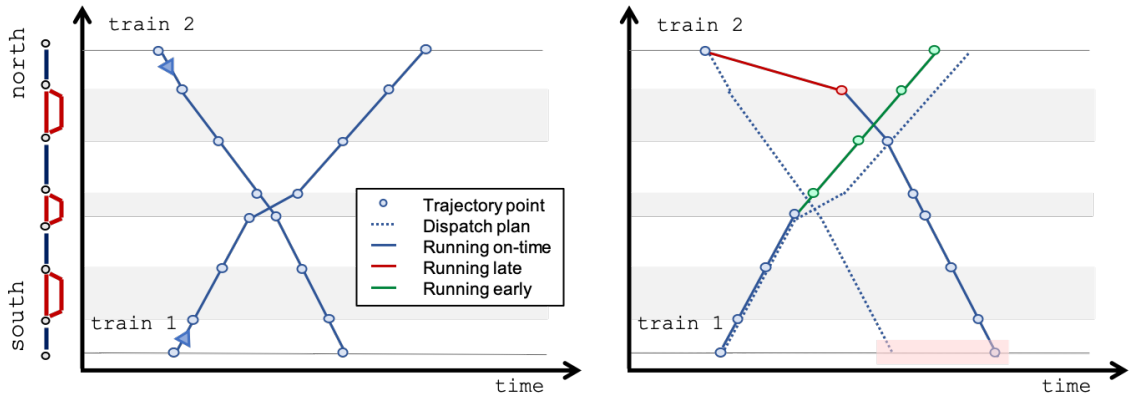


Figure IV.1: Stringline plot of an artificial meet event example, with original plan shown on the left and the resulting train trajectories on the right. The two trains are able to replan their meet location with low impact to overall runtime. An initial delay for train 2, shown by the red section of the trajectory, necessitates replanning.

optimization. They instead rely on heuristic and agent-based approaches. These can work very well in some instances, but are limiting when optimality is core to the research question. Because of this construction, a problem to determine the best of a feasible set of scenarios can consider only a list of candidates, an exhaustive list by brute force, or randomly perturbed scenarios. Quantifying consequences or cost within a simulation is also a limitation because an optimal baseline cannot be computed. Absolute costs are possible, but relativity to a lower bound is not inherent within simulation.

Consider two artificial examples of dispatch replanning in Figures IV.1 and IV.2. Both examples show stringline diagrams for two opposing-direction trains across a small section of track with three sidings and four single-track segments. The planned meet event in both scenarios is depicted at left and the actual dispatch is shown on the right. In the first example (Figure IV.1), the planned meet location between these two trains is at the short middle siding. Train 2 encounters a delay on its first track segment, shown by the red portion of the stringline in the right stringline. The meet event is able to be replanned such that it occurs at the northern siding, instead. Train 1 actually arrives earlier than originally planned, and the added runtime for train 2 is only as large as its initial delay. Conversely, the second example (Figure IV.2) shows a meet event that must occur at one of the longer sidings (north or south) because the trains involved are very long. In this case, an initial delay by train 1 on its first two track segments (shown in red on the right stringline) ends up delaying the meet event significantly because it can not be replanned to the middle siding due to length constraints. This results in a large delay for both trains. These examples demonstrate how train delays are not all created equal because of constraints on replanning given a current network state.

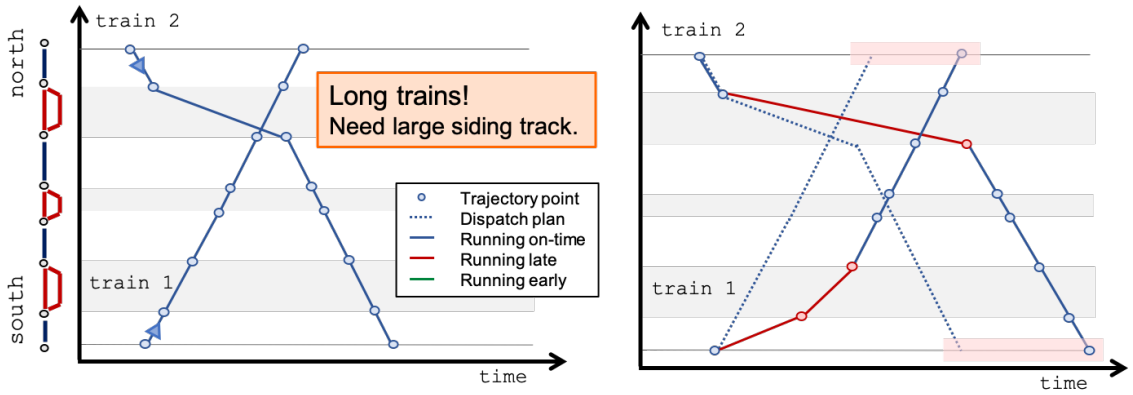


Figure IV.2: Stringline plot of a second artificial meet event example, with original plan shown on the left and the resulting train trajectories on the right. These two trains are not able to replan their meet location because their length constrains the meet event to occur on the northern or southern siding. An initial delay by train 1 results in significant added runtime for both trains because of replanning constraints.

IV.1.2 Problem statement

In this chapter, we introduce an optimization methodology that performs analysis on empirical dispatching data. The methodology considers this historical data in the context of what an optimal decision plan could have been. The method allows one to analyze the decisions that occurred in practice and evaluate how consequential these decisions were. Small train delays and sub-optimal decisions on the railroad are virtually inevitable, and this methodology solves a set of problems that reveal these actions and their effects on the short-term dispatch plan as a whole. We refer to this method as the *dispatch analysis problem* because it can answer a broad set of empirical dispatch analysis questions. In this work we define and address three such problems.

Problem 1: impact of dispatch decisions.

The first question we answer is the following: What is the overall dispatching cost, given the current network state? What events occurred in the evolution of the network state that diminished the ability to continue running the optimal dispatch plan? And how costly is the current network state in terms of its impact on the lower bound dispatch performance moving forward, assuming optimal replanning?

To answer this question, we solve an optimal replanning problem across a period of dispatch time. The beginning of the period is progressed according to the empirical data and we optimally replan for the remainder of the period. The difference between dispatching performance after empirical decisions and replanning versus optimal planning from the beginning is referred to as the *replanned optimality gap*.

Problem 2: alterations to dispatch decisions.

Given a negative effect that the current network state have had on the ability to replan, which specific

alterations could have been made to the network state in order to reduce the replanned optimality gap?

In light of the first question, in the case that costly decisions have been made, it is useful to know whether small changes to decisions in the past could have significantly reduced any negative impacts on core performance criteria. For example, a suboptimal meet location could have not just immediate delay for one of the trains, but also impacts on future meet events that are delayed in turn.

Problem 3: impact of individual trains.

The third addresses specific train in a dispatch: Which trains, in particular, have the largest impact on the ability to run to schedule? And to what degree were these effects caused by the train's own performance or caused by its secondary impact on other trains?

Train volume on many network sections fluctuates over the course of a day or week. As such, trains running during less congested periods could experience large delays, but have very little impact on other trains. Conversely, a train in highly congested periods, or one with which many others interact, can have a large impact on overall dispatch performance even if its own deviation from an optimal dispatch plan is small.

IV.1.3 Contributions

This dispatch analysis problem methodology delivers the ability to critically evaluate empirical dispatching decision making and performance questions. We pose three such dispatch analysis questions and show that each can be posed as a data-constrained optimization problem. We instantiate the specific problem formulation and use it to solve these questions, for which we provide a case study and results.

The three application questions and key findings from the analysis are: 1) Determine immediate future impact of dispatch decisions that have been made; a temporal analysis shows the periods during which replanning optimality gap grew most substantially. 2) Find possible remedies to past decisions that would be particularly impactful; during certain periods, a short list of changes of small magnitude to empirical data was found to have a magnifying effect on potential future performance. 3) Assess the impact that individual trains had on other trains and the dispatch plan as a whole; certain trains exhibit unique behavior in terms of propagation of their own delay onto others and their effects on dispatch decision making.

We are ultimately interested in a variety of dispatch analysis questions. With this methodology, we show that each of these three problems can be posed as a data-contrained optimal dispatch problem and each is a sub-problem of the methodology's general form. Importantly, this methodology (and the three sub-problems) are in the same class of problem as the optimal dispatch problem. In most cases, the dispatch analysis methods could follow directly from the constraint set used in an optimal dispatch model.

The remainder of this chapter is organized as follows. The formulation of the dispatch analysis methodology is explained in Section IV.2. Problem 1, the impact of dispatch decisions, is introduced in Section IV.3.1.

Problem 2, alterations to dispatch decisions, is introduced in Section IV.4. Problem 3, impact of individual trains, is introduced in Section IV.5. The precise instantiation of the optimization model for the dispatch analysis methodology is given in Section IV.6. Section IV.7 discusses data used in the case studies for each problem. Sections IV.8, IV.9, and IV.10 present and discuss results for the three problems. Finally, Section IV.11 concludes the chapter and discusses approaches to future work on the topic.

IV.2 Methodology

IV.2.1 Preliminaries: optimal dispatch problem

In this work, we consider the time values at which trains passed fixed locations on the network. These fixed locations are called *OS-points* and delineate the endpoints of track segments. In this manner, we work with the times at which each train reaches the end of each segment of track. Track segments belong to the ordered set M, where $M : 0, 1, 2, 3, ...$ and is indexed by $m \in M$. Each section of the network has trains running in two directions: directions 1 and 2. Trains in direction 1 are the set I, indexed by $i \in I$, and trains in direction 2 are the set J, indexed $j \in J$. We therefore refer to each timing value, the time at which a train $i \in I$ completed track segment $m \in M$, as $x_{i,m}$. Likewise, for trains $j \in J$, we have $x_{j,m}$.

It is worth noting that when referring to the times at which a train $i \in I$ in direction 1 and a train $j \in J$ cross the same track segment $m \in M$, these times are actually referring to the two separate endpoints of that track segment. Because they operate in opposite directions, the completion points of the segment are at opposite ends. Additionally, note that not all trains operate across every track segment. When relevant, we denote M_i as the subset of track segments on which train $i \in I$ has timing values, where $M_i \subseteq M$; this is the same for trains $j \in J$, where $M_j \subseteq M$.

The collection of time values at OS-points for a train is referred to a train's *trajectory*. Symbolically, we denote the trajectory for train $i \in I$ as $x_{i,m} \forall m \in M_i$; likewise for train $j \in J$: $x_{j,m} \forall m \in M_j$. These trajectories for each train in a dispatch problem are assembled into the vector of decision variables, which we denote x.

Recall that an optimal dispatch problem finds the train trajectories that minimize some measure of dispatching desirability. It may be posed in the general form (discussed in Section III.2.1, but repeated here for convenience):

$$\begin{aligned} \underset{x,z}{\text{minimize:}} \quad & f(x, z) \\ \text{subject to:} \quad & A_1 x + A_2 z \leq b, \end{aligned} \tag{IV.1}$$

where the decision variables are $x \in \mathbb{R}^p_+$ and $z \in \mathbb{Z}^q$. In a common formulation Petersen et al. (1986) (and in this work) Barbour et al. (2018b), the decision variables x encode times at which trains reach various points

on the network, while the integer decision variables z encode dispatching logic that indicates if and where meets and overtakes occur on the network and track assignment for trains. The function f quantifies some measure of dispatching desirability, which is to be minimized.

IV.2.2 Assumptions

We briefly summarize key assumptions that are made in the dispatch analysis methodology and comment on their importance.

Similar to the work on data reconciliation presented in Chapter III, the dispatch analysis methodology considers data at the track segment level. Therefore, the feasibility of a train trajectory is only determined in timing at ends of the segment and minimum train-specific free run times, and not by train performance capabilities in the middle of the segment. This assumption is critical to the particular optimization model presented here, but the methods can be generalized to other higher-fidelity data streams by modifying the model. We must also assume that atypical train movements, such as reversing, did not and can not occur. This is a simplifying assumption, made for model complexity and relative frequency of these movements.

A fundamental assumption with respect to train schedules is that trains were intended to depart at the time where they registered their first OS-point. This is related to a larger point about dependency: there are events outside of the dispatch window, both temporally and spatially, that impacted the trajectories of trains inside the window. A congested dispatch scenario with a high density of train interactions may not have been intended, but rather the result of an upstream delay or late departure caused within a yard. Additionally, events at the beginning or end of a dispatch window should be further analyzed by shifting the window in the relevant direction to capture more context. Given that an optimal dispatching problem across multiple days is computationally difficult, some care must be taken with the dispatch window.

IV.2.3 Dispatch analysis problem general form

We see in the optimal dispatch problem in equation (IV.1) that it finds train trajectories, the time at which each train reaches the end of each track segment, which are denoted x. In the cases where the optimal dispatch problem corresponds to a real scenario that occurred in historical data. While x is a vector of decision variables in the optimization problem, each of these variables also has a corresponding empirical value from the historical data. The decision variable $x_{i,m}$ has a corresponding value that is the true time at which train $i \in I$ reached the end of track segment $m \in M_i$; this empirical value we refer to as $\tilde{x}_{i,m}$.

This dispatch scenario that corresponds to historical data, is delineated temporally between t_{min} and t_{max}, where each are real clock times. Let $\tilde{X}$ be a set of all empirical timing points $\tilde{x}_{i,m}$ and $\tilde{x}_{j,m}$ with values in the interval $[t_{\text{min}}, t_{\text{max}}]$. And, therefore, let X be the set of decision variables in an optimal dispatch problem

(or similar) that corresponds to all the values in the set $\tilde{X}$. That is, an optimal dispatch problem (or similar) must find *optimal* trajectories for all trains across their relevant track segments for which we have empirical data. Also let Z be the set of integer decision variables that correspond to the timing variables in X.

Both of the sets of decision variables, X and Z, are rewritten in their vector form as x and z by ordering variables according to train and track segment.

The most generic form of the dispatch analysis optimization problem minimizes an objective function that is the combination of dispatching desirability, $f(x, z)$ and deviation of dispatched train trajectories from empirical data, $g(x - \tilde{x}, z - \tilde{z})$. This general form may be written:

$$\begin{aligned} \underset{x,z}{\text{minimize:}} \quad & f(x,z) + g(x - \tilde{x}, z - \tilde{z}) \\ \text{subject to:} \quad & A_1 x + A_2 z \leq b \\ & f(x,z) \leq \alpha \\ & g(x - \tilde{x}, z - \tilde{z}) \leq \beta, \end{aligned} \tag{IV.2}$$

where the functions f, quantifying dispatching desirability, and g, quantifying the difference between the decision variables and their empirical values, can be both objective functions and constraints. As constraints, f and g have limit values α and β, respectively. These constraints are included alongside the feasibility constraints for the trajectory values and integer variables: $A_1 x + A_2 z \leq b$.

IV.2.4 Dispatch analysis at a point in time

The constraints on f can be used to require a certain level of dispatch performance, or using g a certain level of agreement with empirical data. These constraints could also be dropped entirely, leaving the optimization of dispatch performance (f), the optimization of agreement with empirical data (g), or a combination of the two.

We now introduce a time parameter τ, which is a clock time in the interval $[t_{\text{min}}, t_{\text{max}}]$. The parameter is used to emulate the delineation of a portion of data which has already happened (before τ) and a portion that has yet to occur (after τ). Note that τ can also be set to equal t_{min} or t_{max}. Let $\tilde{X}_{\tau-}$ denote the subset of $\tilde{X}$, the empirical data, that occurred at or before time τ (i.e., on the interval $[t_{\text{min}}, \tau]$); and let $\tilde{X}_{\tau+}$ be the subset of $\tilde{X}$ that occurred after time τ (i.e., on the interval $(\tau, t_{\text{max}}]$). This separation based on τ is performed based on the empirical data, which has known time values.

The separation of the decision variables, X, is based on the time value of each corresponding empirical point. A decision variable $x_{i,m}$ or $x_{j,m}$ is contained in $X_{\tau-} \subseteq X$ if its corresponding empirical value, $\tilde{x}_{i,m}$ or $\tilde{x}_{j,m}$ has a value less than or equal to τ (i.e., $\tilde{x}_{i,m} \in \tilde{X}_{\tau-}$ or $\tilde{x}_{j,m} \in \tilde{X}_{\tau-}$). Likewise, a decision

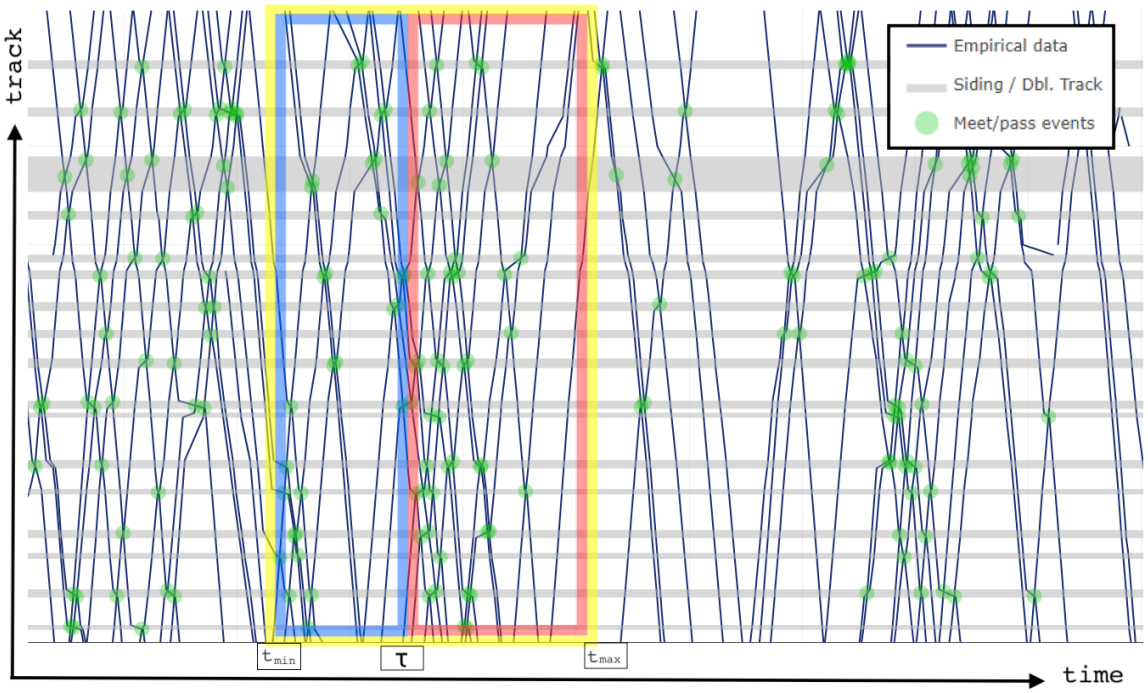


Figure IV.3: Spatiotemporal "stringline" diagram of empirical data, with time on the x-axis and track length along the y-axis. Three points in time are marked: t_{min}, τ, and t_{max}. All train timing points in the yellow box, $[t_{\text{min}}, t_{\text{max}}]$, comprise the set of empirical data $\tilde{X}$. Points inside the blue box, $[t_{\text{min}}, \tau]$ are the subset $\tilde{X}_{\tau-}$; points inside the red box, $[\tau, t_{\text{max}}]$ are the subset $\tilde{X}_{\tau+}$.

variable is contained in $X_{\tau+} \subseteq X$ if its corresponding empirical value is greater than τ (i.e., $\tilde{x}_{i,m} \in \tilde{X}_{\tau+}$ or $\tilde{x}_{j,m} \in \tilde{X}_{\tau+}$). The separation of the set of integer variables, Z, corresponds to sets $X_{\tau-}$ and $X_{\tau+}$. Additionally, we define a division of the sets of trains, I and J, based on τ. Trains that have any portion of their empirical trajectory (i.e., any variable $\tilde{x}_{i,m}$ or $\tilde{x}_{j,m}$, included in the set $\tilde{X}_{\tau-}$ are in sets $I_{\tau-}$ and $J_{\tau-}$. Likewise, trains that have any portion of their empirical trajectory in the set $\tilde{X}_{\tau+}$ are in sets $I_{\tau+}$ and $J_{\tau+}$. It is possible for trains to be included in both sets, respective of their direction. This notation or empirical data and decision variables is summarized in Table IV.1.

Train trajectories are visualized by a time-space diagram called a *stringline*, where the train's speed profile is assumed to be linear between timing points. Figure IV.3 shows an empirical example stringline with the time dimension on the x-axis and the longitudinal spatial dimension of the track on the y-axis. Siding tracks, where trains may meet and pass each other, are shaded grey. Also shown in this figure is a yellow box delineating a time interval $[t_{min}, t_{max}]$, a blue box representing the interval $[t_{\text{min}}, \tau]$ and a red box $[\tau, t_{\text{max}}]$. The portions of the train trajectories inside these boxes would be X (yellow), $X_{\tau-}$ (blue), and $X_{\tau+}$ (red).

Quantity	Description
$\tilde{x}_{i,m}$, $\tilde{x}_{j,m}$	Empirical values for the actual time at which train $i \in I$ or $j \in J$ completed track segment $m \in M$.
$x_{i,m}$, $x_{j,m}$	Individual optimization decision variables for the time at which a train, $i \in I$ or $j \in J$, completed track segment $m \in M$.
t_{min}	Lower time bound of dispatch interval.
t_{max}	Upper time bound of dispatch interval.
τ	Time value in interval $[t_{\text{min}}, t_{\text{max}}]$ that delineates a point at which a specific analysis occurs.
$\tilde{X}$	Set of all empirical timing points that fall within the interval $[t_{\text{min}}, t_{\text{max}}]$.
$\tilde{Z}$	Set of integer values corresponding to $\tilde{X}$.
X	Set of all optimization decision variables corresponding to the empirical values $\tilde{X}$ for the interval $[t_{\text{min}}, t_{\text{max}}]$.
Z	Set of integer variables corresponding to X.
$\tilde{X}_{\tau-}$, $\tilde{Z}_{\tau-}$	Subsets of $\tilde{X}$ and $\tilde{Z}$ where values of $\tilde{x}_{i,m}$ or $\tilde{x}_{j,m}$ are on the interval $[t_{\text{min}}, \tau]$.
$\tilde{X}_{\tau+}$, $\tilde{Z}_{\tau+}$	Subsets of $\tilde{X}$ and $\tilde{Z}$ where values of $\tilde{x}_{i,m}$ or $\tilde{x}_{j,m}$ are on the interval $(\tau, t_{\text{max}}]$.
$X_{\tau-}$, $Z_{\tau-}$	Subsets of X and Z corresponding to values in $\tilde{X}_{\tau-}$ and $\tilde{Z}_{\tau-}$.
$X_{\tau+}$, $Z_{\tau+}$	Subsets of X and Z corresponding to values in $\tilde{X}_{\tau+}$ and $\tilde{Z}_{\tau+}$.
$x_{\tau-}$, $z_{\tau-}$	Ordered vectors of decision variable sets $X_{\tau-}$ and $Z_{\tau-}$ for each train and for each track segment.
$x_{\tau+}$, $z_{\tau+}$	Ordered vectors of decision variable sets $X_{\tau+}$ and $Z_{\tau+}$ for each train and for each track segment.
$\tilde{x}_{\tau-}$, $\tilde{z}_{\tau-}$	Ordered vectors of empirical value sets $\tilde{X}_{\tau-}$ and $\tilde{Z}_{\tau-}$ for each train and for each track segment.
$\tilde{x}_{\tau+}$, $\tilde{z}_{\tau+}$	Ordered vectors of empirical value sets $\tilde{X}_{\tau+}$ and $\tilde{Z}_{\tau+}$ for each train and for each track segment.
$I_{\tau-}$, $J_{\tau-}$	Subsets of trains I and J, which contain trains that have any portion of their empirical trajectory in the set $\tilde{X}_{\tau-}$ (i.e., before or at time τ).
$I_{\tau+}$, $J_{\tau+}$	Subsets of trains I and J, which contain trains that have any portion of their empirical trajectory in the set $\tilde{X}_{\tau+}$ (i.e., after time τ).

Table IV.1: Summary of general notation for optimization variable sets and empirical data.

The general form of the optimization problem based on a separation of data by τ is the following:

$$\begin{aligned}
\underset{x_{\tau-},x_{\tau+},z_{\tau-},z_{\tau+}}{\text{minimize:}} \quad & f_{\tau-}(x_{\tau-},z_{\tau-}) + f_{\tau+}(x_{\tau+},z_{\tau+}) + g_{\tau-}(x_{\tau-}-\tilde{x}_{\tau-},z_{\tau-}-\tilde{z}_{\tau-}) \\
& + g_{\tau+}(x_{\tau+}-\tilde{x}_{\tau+},z_{\tau+}-\tilde{z}_{\tau+}) \\
\text{subject to:} \quad & A_1x + A_2z \leq b \\
& f_{\tau-}(x_{\tau-},z_{\tau-}) \leq \alpha_{\tau-} \\
& f_{\tau+}(x_{\tau+},z_{\tau+}) \leq \alpha_{\tau+} \\
& g_{\tau-}(x_{\tau-}-\tilde{x}_{\tau-},z_{\tau-}-\tilde{z}_{\tau-}) \leq \beta_{\tau-} \\
& g_{\tau+}(x_{\tau+}-\tilde{x}_{\tau+},z_{\tau+}-\tilde{z}_{\tau+}) \leq \beta_{\tau+},
\end{aligned} \tag{IV.3}$$

where a distinction is imposed on functions f and g (originally from equation (IV.2)) according to data in $x_{\tau-}$ or $x_{\tau+}$, resulting in functions $f_{\tau-}$, $f_{\tau+}$, $g_{\tau-}$, and $g_{\tau+}$. The potential constraining values for each of these functions become $\alpha_{\tau-}$, $\alpha_{\tau+}$, $\beta_{\tau-}$, and $\beta_{\tau+}$, respectively. Note that the full vectors of decision variables, x and z, which corresponds to the sets, $X = X_{\tau-} \cup X_{\tau+}$ and $Z = Z_{\tau-} \cup Z_{\tau+}$, are still subject to the operational constraint set that defines a feasible train dispatch.

The interpretation of each function is:

- $f_{\tau-}$: dispatch performance metric evaluated on trajectory points in $x_{\tau-}$, at or before τ.
- $f_{\tau+}$: dispatch performance metric evaluated on trajectory points in $x_{\tau+}$, after τ.
- $g_{\tau-}$: function quantifying the difference between empirical data, $\tilde{x}_{\tau-}$ and $\tilde{z}_{\tau-}$, and corresponding decision variables, $x_{\tau-}$ and $z_{\tau-}$, at or before τ.
- $g_{\tau+}$: function quantifying the difference between empirical data, $\tilde{x}_{\tau-}$ and $\tilde{z}_{\tau-}$, and corresponding decision variables, $x_{\tau+}$ and $z_{\tau+}$, after τ.

The constraint values $\alpha_{\tau-}$, $\alpha_{\tau+}$, $\beta_{\tau-}$, and $\beta_{\tau+}$, are thus interpreted as the upper limit values on each of these functions – the required dispatch performance and the allowable difference between decision variables and empirical data.

IV.3 Problem 1: impact of dispatch decisions

The first application of the dispatch analysis problem is to quantify the impact of the current network state on the ability of the schedule to continue to run at or near optimality. As events happen on the network and trains deviate from the original optimal schedule, the plan must be re-optimized to take into account deviations. We allow the network to evolve up to time τ, and then replanning is initiated based on the positions of trains

at this time. This combination gives us the best possible dispatch achievable given the decisions that have already been made, assuming we make optimal decisions moving forward. It is therefore a lower limit on future performance.

The best possible scenario as time progresses is for trains to maintain the optimal schedule, or to deviate only in such a way that maintains feasibility of the optimal schedule. Any sub-optimal deviation will result in a replanned future schedule that has a larger objective value that the original, optimal schedule. That is, adding additional constraints to the optimal dispatch problem in the form of fixing decision variables to empirical data, can only serve to increase a minimized objective value. As time progresses and the schedule is fixed further to the empirical data, the objective value of the replanned schedule will converge on the objective value of the wholly empirical data (i.e., when τ reaches the end of the data window $[t_{\text{min}}, t_{\text{max}}]$). The specific steps in this analysis are as follows:

1. For a time window $[t_{\text{min}}, t_{\text{max}}]$, solve the optimal dispatch problem (equation (IV.1)) given only the initial condition of each train. This establishes the **baseline** dispatch where we refer to the total runtime of all trains as r_0.

2. Assemble the empirical trajectories of all trains within the interval $[t_{\text{min}}, \tau]$: $\tilde{x}_{\tau-}$.

3. Fix the decision variables $x_{\tau-}$ to their empirical values, $\tilde{x}_{\tau-}$.

4. Solve the new optimization problem with the added constraints, given in equation (IV.4).

5. The new objective value is the best dispatch achievable given the decisions made up to τ. The total runtime of all trains for this problem is denoted r_τ, where $r_\tau \geq r_0$.

IV.3.1 Problem 1 formulation

In this formulation we wish to find the best dispatch plan, while before time τ holding the difference between each empirical data point and corresponding timing variable, $g_{\tau-}(x_{\tau-} - \tilde{x}_{\tau-}, z_{\tau-} - \tilde{z}_{\tau-})$, to zero. This fits into the general dispatch analysis formulation as:

$$\begin{aligned} \underset{x,z}{\text{minimize:}} \quad & f(x,z) \\ \text{subject to:} \quad & A_1 x + A_2 z \leq b \\ & g_{\tau-}(x_{\tau-} - \tilde{x}_{\tau-}, z_{\tau-} - \tilde{z}_{\tau-}) = 0, \end{aligned} \tag{IV.4}$$

In this problem, we take the objective function f, quantifying the dispatch performance, to be the sum of the runtime of all trains. The distribution of any delay on the route is not of concern, so long as the train

reaches its destination at the earliest possible time. The form of f is therefore:

$$f(x,z) = \sum_{i \in I} x_{i,q_i} + \sum_{j \in J} x_{j,q_j}, \tag{IV.5}$$

where q_i and q_j are the final track segments of the trajectories for trains $i \in I$ and $j \in J$, respectively. This means that x_{i,q_i} and x_{j,q_j} denote the completion time of the final track segment for trains i and j.

In order to hold the value of each timing variable in $x_{\tau-}$ to its empirical value in $\tilde{x}_{\tau-}$, we impose the $\mathcal{L}_1$ norm on the difference between decision the variables and constraint the value of $g_{\tau-}$ to be zero:

$$g_{\tau-}(x_{\tau-} - \tilde{x}_{\tau-}, z_{\tau-} - \tilde{z}_{\tau-}) = ||x_{\tau-} - \tilde{x}_{\tau-}||_1, \tag{IV.6}$$

where $||\cdot||_1$ is the $\mathcal{L}_1$ norm.

IV.4 Problem 2: alterations to dispatch decisions

We presented in Section IV.3 a method to study the impact of empirical decision making on the baseline optimal dispatch plan. This reveals the temporal manner in which deviations from the baseline plan impacted total train runtime. We are now interested in isolating more specific instances of empirical performance that could have been *changed* in order to improve future dispatch performance. At a given time τ, where the empirical data before τ, $\tilde{x}_{\tau-}$, has introduced Δ minutes of additional runtime to the dispatch in excess of the baseline optimal dispatch, we find the minimal changes to the empirical data in $\tilde{x}_{\tau-}$ that could be made which would decrease Δ to a desired level.

The steps to address this problem are as follows:

1. For a time window $[t_{\text{min}}, t_{\text{max}}]$, solve the optimal dispatch problem (equation (IV.1)) given only the initial condition of each train. This establishes the **baseline** dispatch where we refer to the total runtime of all trains as r_0.

2. Assemble the empirical trajectories of all trains within the interval $[t_{\text{min}}, \tau]$: $\tilde{x}_{\tau-}$.

3. Fix the decision variables $x_{\tau-}$ to their empirical values, $\tilde{x}_{\tau-}$.

4. Solve the new optimization problem with the added constraints, given in equation (IV.4), which is the best dispatch achievable given the decisions made up to τ. The total runtime of all trains for this problem is denoted r_τ, where $r_\tau \geq r_0$.

5. Calculate $\Delta = r_\tau - r_0$, the runtime in excess of the baseline dispatch that was added because of the empirical decisions up to τ. Determine a reduced value of Δ, which is to be achieved by modifying

the empirical data; we denote this value Δ' and calculate the desired total runtime as $r' = r_0 + \Delta'$.

6. Impose a constraint on the total runtime, which must be less than or equal to r'.

7. Solve the empirical improvement problem, which minimizes the alteration to the empirical data before τ, $\tilde{x}_{\tau-}$, while achieving a total runtime of all trains less than or equal to r'. This problem is given by equation (IV.7).

IV.4.1 Problem 2 formulation

This problem minimizes the alteration to the empirical data, $g_{\tau-}(x_{\tau-} - \tilde{x}_{\tau-}, z_{\tau-} - \tilde{z}_{\tau-})$, before τ that is required to reduce the objective value of the replanned dispatch, $f(x, z)$, at τ to a desired level, α:

$$\begin{aligned} \underset{x,z}{\text{minimize:}} \quad & g_{\tau-}(x_{\tau-} - \tilde{x}_{\tau-}, z_{\tau-} - \tilde{z}_{\tau-}) \\ \text{subject to:} \quad & A_1 x + A_2 z \leq b \\ & f(x, z) \leq r', \end{aligned} \tag{IV.7}$$

where r' is the value of runtime of all trains that is to be achieved by alteration of empirical data; it is described above in step 5.

We again define $f(x, z)$, the performance function for dispatched trajectories that is constrained to reduce the lower bound of the replanned dispatch, to be the total runtime of all trains:

$$f(x, z) = \sum_{i \in I} x_{i,q_i} + \sum_{j \in J} x_{j,q_j}, \tag{IV.8}$$

where q_i and q_j are the final track segments of the trajectories for trains $i \in I$ and $j \in J$, respectively, and x_{i,q_i} and x_{j,q_j} denote the completion time of the final track segment for trains i and j.

We define the objective function $g_{\tau-}$, the alteration to empirical trajectories before τ, as:

$$\begin{aligned} g_{\tau-}(x_{\tau-} - \tilde{x}_{\tau-}, z_{\tau-} - \tilde{z}_{\tau-}) = & \sum_{i \in I_{\tau-}} \sum_{n=p_i+1}^{q_i} |(x_{i,n} - x_{i,n-1}) - (\tilde{x}_{i,n} - \tilde{x}_{i,n-1})| \\ & + \sum_{j \in J_{\tau-}} \sum_{n=p_j-1}^{q_i} |(x_{j,n} - x_{j,n+1}) - (\tilde{x}_{j,n} - \tilde{x}_{j,n+1})|, \end{aligned} \tag{IV.9}$$

where $I_{\tau-}$ and $J_{\tau-}$ are the subsets of I and J for which some portion of the trajectory of $i \in I$ or $j \in J$ are included in $X_{\tau-}$; p_i and p_j are the first track segments in the trajectories of i and j that are included in $X_{\tau-}$; q_i and q_j are the final track segments in the trajectories of i and j that are included in $x_{\tau-}$. This function computes the summation of the differences in segment runtimes between the decision variables $x_{\tau-}$ and their

empirical values $\tilde{x}_{\tau-}$. The differences in segment runtimes are used (as opposed to the difference in timing points) because any alteration applied to the difference between these values will affect each successive timing point in the trajectory, effectively shifting the remainder of the train's trajectory in time. This is analogous to altering a train's empirical trajectory so that it would have run faster on a particular track segment, and thus arrived at successive OS-points sooner.

IV.5 Problem 3: impact of individual trains on dispatch plan

Quantifying the impact of empirical decisions run up to a given time, described in Problem 1 in Section IV.3, provided information on a temporal basis. It can also be informative to assess the empirical data with respect to specific trains. Beyond the impact of runtime incurred by a train itself, what impact did that train have on others in the dispatch plan?

In this problem, we find not the alterations to a train's trajectory, but instead the impact that fixing such train's trajectory has on the rest of the schedule. Small deviations of a train's trajectory from the optimal schedule have the potential to impact other train's significantly if it was tightly integrated, indicating that the schedule is very sensitive to that train. Conversely, a train may have very little secondary effect on the schedule, even if it experiences large deviations from its optimal trajectory.

This problem is similar to the quantification of knock-on delay, but more precisely, it measures the secondary impact on the optimal schedule from a train's deviation in its own optimal trajectory. The steps to evaluating a specific train, denoted w, in this application are as follows:

1. For a time window $[t_{\text{min}}, t_{\text{max}}]$, solve the optimal dispatch problem (equation (IV.1)) given only the initial condition of each train. This establishes the **baseline** dispatch, where we refer to the total runtime of all trains as r_0

2. Let the runtime of train w in the baseline dispatch be γ_w.

3. Assemble the empirical trajectory of train w from all empirical data, $\tilde{X}$, on the interval $[t_{\text{min}}, t_{\text{max}}]$. Let the empirical runtime of train w be γ'_w.

4. Fix the decision variables $x_{w,m}$ to their empirical values $\tilde{x}_{w,m}$ for all tracks segments in the trajectory of train w.

5. Solve the new optimization problem with the added constraints, given in equation (IV.10), which is the best dispatch achievable given fixed trajectory of train w. The total runtime of all trains with train w fixed is denoted r_w, where $r_w \geq r_0$.

(a) The difference between the empirical runtime of train w and its baseline dispatch value, $\gamma'_w - \gamma$, is the *primary added runtime* for this train.

(b) The difference between the runtime with train w fixed and the baseline runtime, for all trains except w, is the *secondary added runtime*. This can be calculated by subtracting the runtime difference for w from the overall runtime difference: $(r_w - r_0) - (\gamma'_w - \gamma_w)$.

IV.5.1 Problem 3 formulation

In this problem, we wish to find the best dispatch of all trains, $f(x, z)$, but with the variables for train w fixed to their empirical values by holding function $g(x_w - \tilde{x}_w, z_w - \tilde{z}_w)$ equal to zero:

$$\begin{aligned} \underset{x,z}{\text{minimize:}} \quad & f(x, z) \\ \text{subject to:} \quad & A_1 x + A_2 z \leq b \\ & g(x_w - \tilde{x}_w, z_w - \tilde{z}_w) = 0, \end{aligned} \tag{IV.10}$$

where w is a specific train in the dataset that is being assessed, x_w and z_w are the decision variables for train w, and $\tilde{x}_w$ and $\tilde{z}_w$ are the empirical values for train w.

We again define $f(x, z)$, the objective value for dispatched trajectories, to be the total runtime of all trains:

$$f(x, z) = \sum_{i \in I} x_{i,q_i} + \sum_{j \in J} x_{j,q_j}, \tag{IV.11}$$

where q_i and q_j are the final track segments of the trajectories for trains $i \in I$ and $j \in J$, respectively, and x_{i,q_i} and x_{j,q_j} denote the completion time of the final track segment for trains i and j. Note that minimizing equation (IV.11) is an equivalent optimization to minimizing the runtime of all trains excluding train w when the trajectory for w is fixed, but all trains are included in $f(x, z)$ for simplicity. This just requires a separation of runtime by train from the value of f.

In order to fix the empirical trajectory of train w, we impose for function g an $\mathcal{L}_1$ norm on the difference between empirical and solved trajectory timing values of train w, and constrain the value of g to be zero:

$$g(x_w - \tilde{x}_w, z_w - \tilde{z}_w) = ||x_w - \tilde{x}_w||_1, \tag{IV.12}$$

where $|| \cdot ||_1$ is the $\mathcal{L}_1$ norm, x_w refers to the trajectory timing variables for train w, and $\tilde{x}_w$ refers to its empirical trajectory timing values.

Quantity	Symbol	Details
Set of main line tracks	$M : 0, 1, 2, 3, ...$	Track indices begin at zero and increase in direction 1.
Set of siding tracks	S	Siding tracks are a subset of main line tracks $S \subset M$: those that contain more than one track for meet and pass maneuvers.
Track length	K_m	Length of each track segment $m \in M$.
Trains in direction 1	I	Individual trains are denoted $i \in I$.
Trains in direction 2	J	Individual trains are denoted $j \in J$.
Train length	L_i and L_j	Length of each train in the same units as K_m.
Empirical track segment completion time	$\tilde{x}_{i,m}$, $\tilde{x}_{j,m}$	Time at which train i or train j reached the end of segment m, as indicated in empirical data. *Note that because of directionality, these variables for trains i and j refer to opposite sides of the track segment m.*
Main line free run traversal time	$T_{i,m}$ and $T_{j,m}$	Minimum traversal time of segment m for train i (or j) if the train is on the main line track.
Siding free run traversal time	$U_{i,s}$ and $U_{j,s}$	Minimum traversal time of siding segment s for train i (or j).
Clearance headway, opposing direction	$H_{m,i,j}$	Minimum clearance time separation of trains in opposing directions, dependent on track segment.
Follow headway, same direction	H_{m,i_1,i_2} and H_{m,j_1,j_2}	Minimum following time separation of trains in the same direction, dependent on track segment.

Table IV.2: Optimization model parameters for the dispatch analysis problem.

IV.6 Instantiation of the dispatch analysis problem

IV.6.1 Parameters and variables

The parameters and variables for the instantiation of the dispatch analysis problem are identical to those of the data reconciliation problem. They are repeated briefly in Table IV.2 and Table IV.3 for reference. The full explanation of this notation can be found in Section III.3.

IV.6.2 Constraints

The constraint set for the dispatch analysis problem, abbreviated in equations (IV.2) and (IV.3) as $A_1 x + A_2 z \leq b$, is also identical to that of the data reconciliation problem discussed in Section III.3. The constraints are shown in Table IV.4 for reference, but discussed in full only in Section III.3. Constraints are given in logical form using `IF` and `THEN` statements, but in actual implementation are encoded in mixed-integer form.

IV.6.3 Objective function

The objective functions considered for the dispatch analysis problem at present are unique to each analysis problem. Problem 1 minimizes overall runtime of trains; though, with all decision variables up to a point τ fixed to their empirical values, it effectively minimizes future runtime of trains. Problem 2 minimizes

Quantity	Symbol	Details
Track segment completion time	$x_{i,m}$, $x_{j,m}$	Decision variable for the time at which train i or train j reaches the end of segment m. These correspond to the empirical timing values $\tilde{x}_{i,m}$ and $\tilde{x}_{j,m}$. *Note that because of directionality, these variables for trains i and j refer to opposite sides of the track segment m.*
Train ordering (opposing direction)	$\pi_{i,j,m}$	Denotes which train crossed segment $m \in (M \backslash S)$ first: 1 if train i and 0 if train j.
Train ordering (same direction)	$\phi_{i_1,i_2,m}$ and $\phi_{j_1,j_2,m}$	Denotes which train crossed segment $m \in M$ first: 1 if i_1 and 0 if i_2, where $i_1 \neq i_2$.
Meet events	$\mu_{i,j,s}$	Denotes whether a meet event occurred between trains i and j at siding s: 1 if true and 0 if false.
Overtake events	$\rho_{i_1,i_2,s}$ and $\rho_{j_1,j_2,s}$	Denotes whether train i_1 overtook train i_2 on siding s (or j_1 overtook j_2): 1 if true and 0 if false.
Siding assignment	$\sigma_{i,s}$ and $\sigma_{j,s}$	Indicates whether train i (or j) took the siding track s instead of the main line track: 1 if true and 0 if false.

Table IV.3: Optimization model decision variables for the dispatch analysis problem.

the $\mathcal{L}_1$ norm of the differences between segment runtime of the empirical trajectories and decision variable trajectories. Problem 3 minimizes the overall runtime of trains; with the trajectory for a specific train, w, fixed to its empirical values, it effectively minimizes runtime of all trains except w. The exact form of these objective functions are given in equations (IV.5), (IV.9), and (IV.11).

IV.7 Case study data preparation

We now present a description of freight rail dispatch data that is used as a case study to answer the three dispatch analysis questions in this work.

The historical dispatch dataset is collected from a rail network section of a U.S. Class-I railroad, between Nashville, Tennessee, and Birmingham, Alabama. The network section is single track with 17 passing sidings of various lengths and a total of 37 track segments. It is approximately 190 miles (305 km) in length. The area and track are shown in Figure IV.4, highlighted by the yellow area. Single-track sections are shown in blue and sidings are in red. Nashville, TN, is at the north end of the network section and Birmingham, AL, to the south.

Case study analyses are given for various ranges of data: those isolating a single window of dispatching data are taken from nine hours of data during the week of January 4, 2016; the exact time window is not given for data confidentiality reasons. Analyses are extended beyond the detailed discussion of the single nine-hour window; multiple windows of dispatching data from January 1, 2016, to January 31, 2016 are considered and their results aggregated. All dispatch data is reconciled according to the process developed and discussed in Chapter III in order to remove any small errors or omissions and ensure feasibility prior to use in the dispatch analysis problem. The reconciliation process also generates, for each window of dispatch data, the relevant

Constraint	Equation	Details
Travel time, direction 1	$x_{i,m} \geq x_{i,m-1} + T_{i,m}$	Travel time on segment m, calculated $x_{i,m} - x_{i,m-1}$, must be at least $T_{i,m}$.
Travel time, direction 2	$x_{j,m} \geq x_{j,m+1} + T_{j,m}$	Travel time on segment m, must be at least $T_{j,m}$. *Note: segment ordering in direction 2 is opposite that of direction 1. Segment $m+1$ is the segment prior to segment m in direction 2.*
Siding travel time, direction 1	IF $\sigma_{i,s} = 1$, THEN $x_{i,s} \geq x_{i,s-1} + U_{i,s}$	If train i took siding s, then it must have minimum travel time $U_{i,s}$.
Siding travel time, direction 2	IF $\sigma_{j,s} = 1$, THEN $x_{j,s} \geq x_{j,s+1} + U_{j,s}$	If train j took siding s, then it must have minimum travel time $U_{j,s}$.
Opposite direction clearance	IF $\pi_{i,j,m} = 1$, THEN $x_{i,m} + H_{i,j} \leq x_{j,m+1}$, ELSE $x_{j,m} + H_{i,j} \leq x_{i,m-1}$	If trains i completed siding s before j, then impose clearance headway $H_{i,j}$ on their timing.
Clearance headway, direction 1	IF $\phi_{i_1,i_2,m} = 1$, THEN $x_{i_1,m} + H_{i_1,i_2} \leq x_{i_2,m}$	If train i_1 completed track $m \in M$ before i_2, then impose clearance headway H_{i_1,i_2} on their timing.
Clearance headway, direction 2	IF $\phi_{j_1,j_2,m} = 1$, THEN $x_{j_1,m} + H_{j_1,j_2} \leq x_{j_2,m}$	If train j_1 completed track $m \in M$ before j_2, then impose clearance headway H_{j_1,j_2} on their timing.
Single track ordering, direction 1	$\phi_{i_1,i_2,m} = \phi_{i_1,i_2,m-1}$	Trains i_1 and i_2 must maintain the same order across all single tracks $m \in (M \setminus S)$.
Single track ordering, direction 2	$\phi_{j_1,j_2,m} = \phi_{j_1,j_2,m+1}$	Trains j_1 and j_2 must maintain the same order across all single tracks $m \in (M \setminus S)$.
Simultaneous meet/overtake, direction 1	$\sum_{j \in J} \mu_{i_1,j,s} + \sum_{i_2 \in (I \setminus \{i_1\})} \rho_{i_1,i_2,s} \leq 1$	Restrict the number of meet and overtake events involving train i_1 to one at siding s.
Simultaneous meet/overtake, direction 2	$\sum_{i \in I} \mu_{i,j_1,s} + \sum_{j_2 \in (J \setminus \{j_1\})} \rho_{j_1,j_2,s} \leq 1$	Restrict the number of meet and overtake events involving train j_1 to one at siding s.
Meet siding assignment	IF $\mu_{i,j,s} = 1$, THEN $\sigma_{i,s} + \sigma_{j,s} = 1$	If trains i and j met at siding s, require one of them to take the siding track.
Overtake siding assignment, direction 1	IF $\rho_{i_1,i_2,s} = 1$, THEN $\sigma_{i_1,s} + \sigma_{i_2,s} = 1$	If train i_1 overtakes i_2 at siding s, require one of them to take the siding track.
Overtake siding assignment, direction 2	IF $\rho_{j_1,j_2,s} = 1$, THEN $\sigma_{j_1,s} + \sigma_{j_2,s} = 1$	If train j_1 overtakes j_2 at siding s, require one of them to take the siding track.
Siding length, direction 1	IF $L_i > K_s$, THEN $\sigma_{i,s} = 0$	Train i may not use siding track s if it is too long.
Siding length, direction 2	IF $L_j > K_s$, THEN $\sigma_{j,s} = 0$	Train j may not use siding track s if it is too long.

Table IV.4: Listing of logical and operational constraints for the dispatch analysis problem.

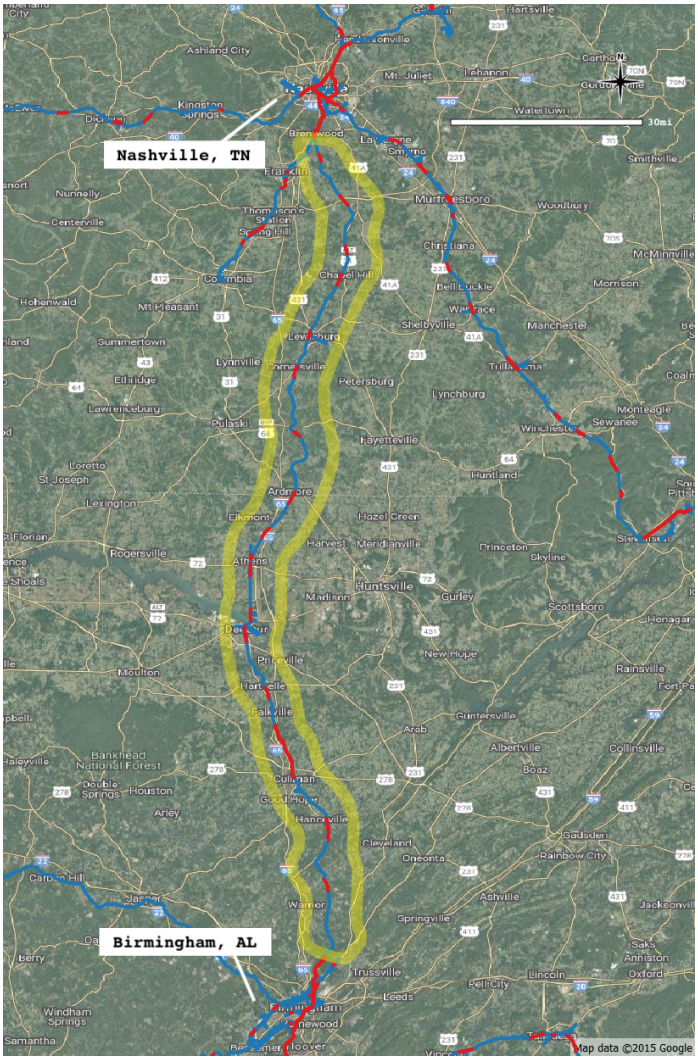


Figure IV.4: Depiction of the 190-mile study area between Nashville, TN, and Birmingham, AL. The section is predominantly single track (blue) with 17 passing sidings, shown in red.

objective function values for the purely empirical data.

Values are derived via historical data mining for the following optimization parameters: directional minimum main line track runtime ($T_{i,m}$ and $T_{j,m}$), directional minimum siding track runtime ($U_{i,s}$ and $U_{j,s}$), clearance headway for meet events specific to each end of each siding ($H_{m,i,j}$), and follow headway for same direction trains specific to each direction and each track segment (H_{m,i_1,i_2} and H_{m,j_1,j_2}).

Runtime distributions for each track segment in each direction were mined from two years of historical data: January 2014 to December 2015. The minimum main line free run traversal time of each track segment in each direction, $T_{i,m}$ and $T_{j,m}$, was taken to be the 90th percentile lowest observed value for each direction. This choice was made based on inspection of runtime distributions. Historical timing at OS-points in the dataset is given to the nearest minute, which results in unreasonably low runtime value on short track segments if the pure minimum value is used (e.g., 1 minute on a 1.5-mile track segment, implying 90mph travel speed). Applying the 90th percentile rule mitigates this rounding effect without requiring deviation of a significant number of trains (or a large magnitude of deviation) when data reconciliation is performed, due to their lower runtime values on segments.

In order to ascertain siding track traversal times, the runtime values for trains during meet events were isolated. Within each meet event, the main line runtime was assumed to be the lower of the two values and the siding runtime was assume to be the higher of the two. The two values for each meet event were also separated by direction, resulting in four set of values: lower runtimes in direction 1, higher runtimes in direction 1, and lower/higher in direction 2. Figure IV.5 shows these lower and higher runtime value distributions for direction 1 trains taken from meet events, across a subset of sidings on the network section. Three plots, grouped by color for each siding, show the set of lower runtime values (left), the set of higher runtime values (right), and the comparison set of all observed runtimes regardless of whether a meet occurred. Each violin plot is similar to a histogram, with the widest area indicating a higher frequency of values than the thinner tails. The median value is marked with the solid dash and one standard deviation to each side is shown with the dotted segments. The scale for exact runtime values is not given for data confidentiality, but clear distinction in the lower/higher runtime values for meet events can be seen for each siding. In actuality, trains are sometimes stopped on the mainline to allow the meeting train to continue moving on the siding track; for the architecture of this dispatch model, though, the larger minimum runtime is imposed for the train assigned to the siding track.

The 90th percentile smallest runtime value is taken from the set of higher runtimes for each direction to be the minimum siding runtime for the optimization parameters $U_{i,s}$ and $U_{j,s}$ (direction 1 and direction 2, respectively).

Headway values for trains in opposing directions, relevant for meet event clearance time, is similarly

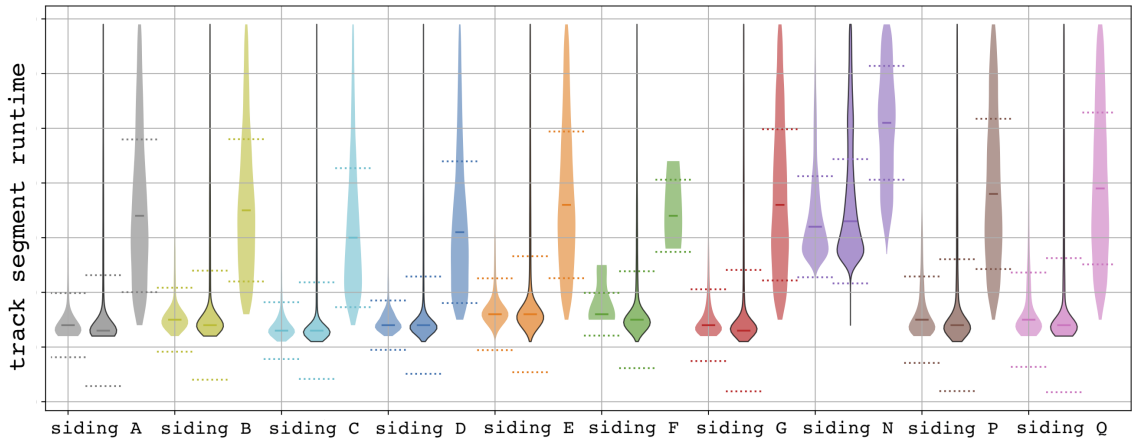


Figure IV.5: Distribution of runtimes on select sidings during meet events. For each siding, grouped by color, the left violin plot is the distribution of runtimes for the faster of the two trains in the meet event and the right plot is the distribution of runtimes for the slower of the two trains. The middle plot in each group, outlined in black, is the distribution of all trains on the siding, not just those in meet events.

taken from the set of meet events observed to occur at each siding. Each endpoint of the siding track is considered separately by computing the difference in arrival time at this point by the two opposing-direction trains. The 80th percentile value is taken as the headway value in order to reduce rounding effects that result in very small headway values at certain sidings.

For same-direction headway, we consider all trains in the two-year mined dataset at the end of each track segment, independently. These trains are sorted by their arrival time at the end of the track segment and the different between each successive pair of trains is computed. Those with separation times of less than 30 minutes are assumed to be roughly following each other, and from this filtered set of separation times we take the 95th percentile minimum value as the minimum follow headway. The 95th percentile threshold was chosen based on manual inspection to reduce the effects of rounding, but differs from opposing direction headway because in these cases many successive pairs of trains were not following as closely, resulting more large values for separation time.

IV.8 Problem 1: results

The analysis question we set out to answer in problem 1 is how to quantify the cost of past dispatch decisions on replanning in the future. As discussed in Section IV.3, we fix a set of empirical dispatching data on the time interval $[t_{\min}, \tau]$ and then, considering the true locations of trains at time τ, optimally replan into the future, $(\tau, t_{\max}]$. The objective function used for dispatching is the minimization of overall train runtime. The interpretation of an objective function value at time τ is the lower bound on total train runtime if optimal decisions are made from τ forward. We first show results on applying this analysis to a single dispatch

window, and then expand the analysis to cover two weeks of dispatching.

Consider the stringline diagram in Figure IV.6. Again, time is on the x-axis, track position is on the y-axis, and sidings are shaded in grey along the track dimension. This diagram shows a portion of the optimal dispatch plan in green alongside the empirical dispatch data for this time interval, in blue. Many of the empirical trajectories can be seen diverging from their optimal trajectories. When we fix empirical data up to a time τ, we then assume that trains are at their empirical locations at this time and must be dispatch from there. Because many of the empirical trajectories diverge, significant replanning must be performed in order to make a new optimal plan.

Figure IV.7 shows the replanning process at time $\tau = 300$, which is marked on the stringline diagram by the dashed blue line. Empirical trajectories (dark blue) are run up to their last observed timing point before $\tau = 300$, which is why some stop short of the dashed blue demarcation. At this point, they have diverged significantly from what could have been their optimal trajectories up to this point, shown in green. After $\tau = 300$, replanning must be performed to develop the new plan; these replanned trajectories are shown in red. The total runtime of the empirical trajectories (blue) plus the replanned trajectories (red) make up the total train runtime for this replanned dispatch at time $\tau = 300$.

Each time more empirical data is introduced into the optimal dispatch problem the value of the objective function (total train runtime) must either remain the same (if optimal decisions were made) or increase (if any sub-optimal decisions were made). The empirical data adds not only its own sub-optimality with respect to the baseline optimal dispatch plan (i.e., in the form of train delay or alterations to the optimal plan), but it also has the potential secondary effect of requiring a change to the future of the optimal plan. The replanned future is optimal given the constraints, but introduces cost because of the sub-optimal positions of trains at the replanning point. The separation of these primary and secondary effects is not addressed in this work.

IV.8.1 Analysis on a single dispatch period

We simulate the effect of dispatching moving forward in time by gradually increasing the τ parameter in 30-minute increments across a 9-hour dispatch time window, from 0 minutes to 540 minutes. The dispatch time window is described in Section IV.7. Indeed, increasing the τ parameter increases the overall objective function value as can be seen in Figure IV.8. The green line shows the lower bound objective value (total train runtime), considering the empirical data and the replanned future. As previously mentioned for Figure IV.7, the total runtime of empirical trajectories (blue) plus the replanned future (red) constitute this total runtime lower bound at a specific τ value (a point on the green curve in Figure IV.8). At $\tau = 0$, no dispatching decisions have yet occurred that will introduce sub-optimality; therefore, no cost is incurred by replanning and the dispatch is effectively the baseline plan. This baseline runtime is marked by the light blue dashed line

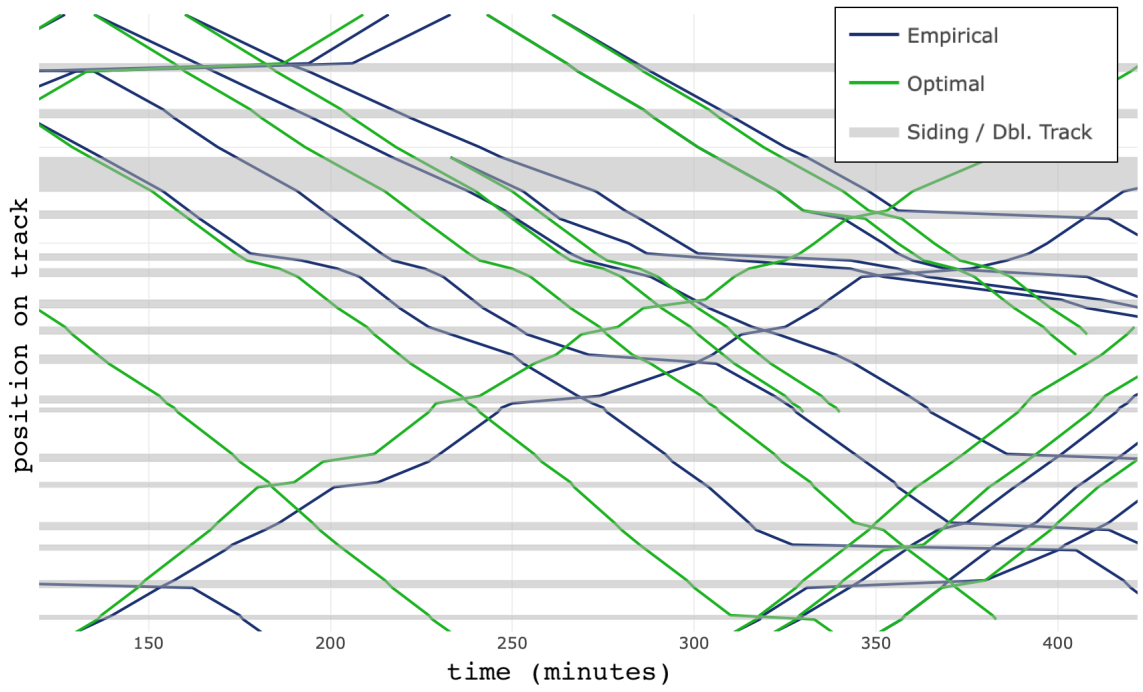


Figure IV.6: Stringline diagram of optimal baseline dispatch plan (shown in green) versus empirical dispatch (dark blue).

in Figure IV.8. At $\tau = 540$, the end of the dispatch window, all empirical dispatch decisions have occurred and no replanning is performed; therefore, the cost of replanning will be the empirical total runtime. This runtime is marked by the grey dashed line.

At the point $\tau = 300$ in Figure IV.8, the plan for which we visualized in Figure IV.7, the lower bound runtime now has a value of $r_{300} = 3178$ minutes, compared with the optimal value of $r_0 = 2549$ minutes. However, the increase in lower bound runtime from the previous point, $\tau = 270$, is modest: $r_{300} - r_{270} = 72$ minutes. Compare this to the increase in lower bound from $\tau = 120$ to $\tau = 150$ or the increase from $\tau = 390$ to $\tau = 420$, which are much more severe: $r_{150} - r_{120} = 164$ minutes and $r_{420} - r_{390} = 286$ minutes. This indicates that decisions made on these latter time intervals, $[120, 150]$ and $[390, 420]$ were much more costly to the dispatch plan. Indeed, Figure IV.9 shows the amount of increase in the lower bound runtime over each successive step of τ; the intervals mentioned experience the largest increases in lower bound runtime over this dispatch window. Again, the separation of whether this was due to primary or secondary effects is a separate question, but it indicates where, temporally, runtime is being introduced in excess of what is possible in the optimal case.

Figure IV.10 depicts multiple plots simultaneously, as they have evolved up to $\tau = 300$. The stringline diagram shows evolved network state up to $\tau = 300$ in blue and future replanned trajectories in red. The

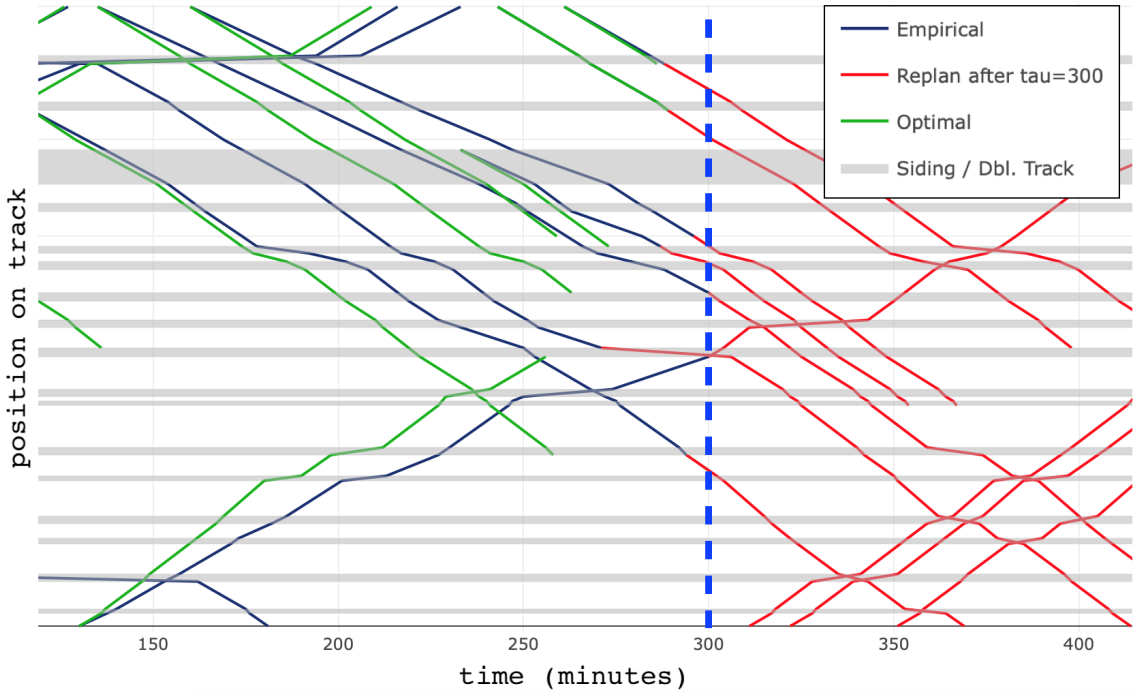


Figure IV.7: Stringline diagram for the same dispatch scenario as Figure IV.6, with dispatch replanning at $\tau = 300$. Empirical data is shown by the dark blue trajectories up to time $\tau = 300$ (marked by the dashed blue line). The trajectories of trains under the baseline (optimal) plan are shown by the green trajectories, for comparison of the empirical versus optimal locations of the trains at time τ. The replanned trajectories moving forward from the empirical locations at $\tau = 300$ are in red. The sections of empirical (blue) plus the replanned (red) trajectories constitute the total runtime value that is evaluated in this section.

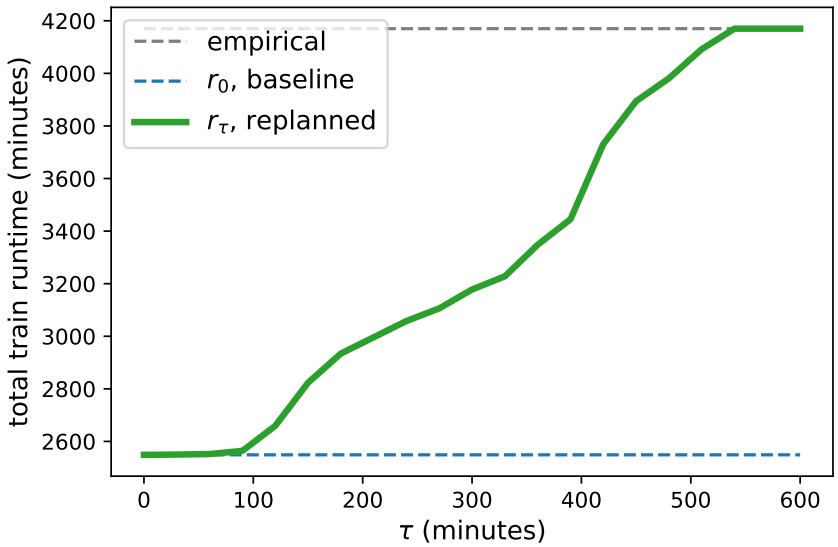


Figure IV.8: Lower bound on train runtime, r_τ, at time τ, when empirical decisions are run from $t_{\min}$ to τ and the optimally-replanned dispatch is executed from τ to $t_{\max}$. As additional empirical decisions are taken into account, the lower bound runtime increases from the baseline optimal value ($r_0 = 2549$ minutes) when $\tau = 0$ to the empirical value (4170 minutes) after $\tau = 540$.

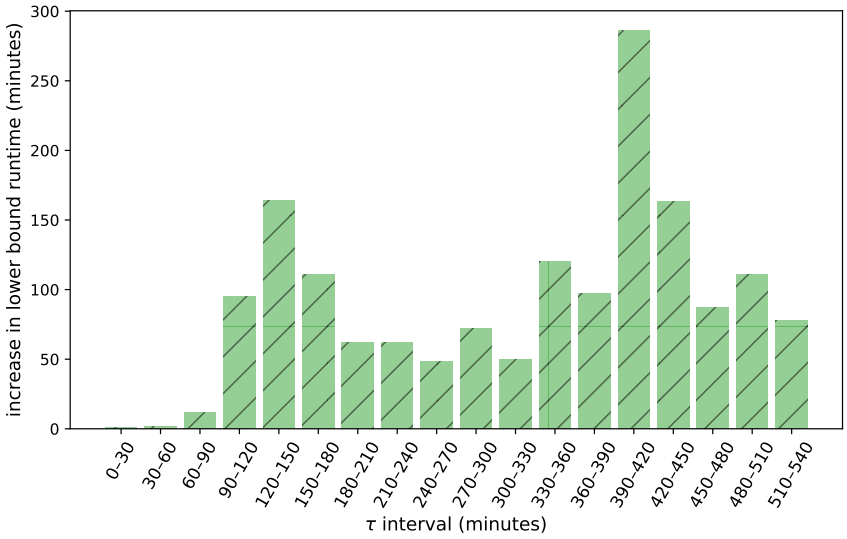


Figure IV.9: Increase in lower bound runtime caused by each timestep of τ. Larger bars indicate larger increases in the lower bound and, thus, a costly change in the network state over the respective time interval.

lower portions of the figure (left to right) show, up to τ, the status of trains in the dispatch window, the evolution of the replanned runtime lower bound, and the increase in lower bound during each time interval.

IV.8.2 Results across multiple periods

The trend in how the lower bound runtime increases across each particular window of dispatch data is expected to differ. The trend will be affected not only by the dispatching decisions but also by the distribution of trains on the network segment. Figure IV.11 shows the accumulation patterns of runtime across two weeks of data in 9-hour windows shifted by 3 hours (thereby overlapping by 6 hours). For comparison purposes, the replanned lower bound values were min-max normalized to [0, 1] using the baseline and empirical runtime values. As is to be expected, there is fairly wide variation in how each accumulates delay within the 9 hours. The mean trend is shown with the black dashed line.

Two particular 9-hour dispatch time windows are highlighted in purple, labeled "morning A" and "morning B". These two are highlighted because they demonstrate very opposing trends in how their lower bound runtimes evolved. By $\tau = 90$ minutes, the lower bound runtime on "morning A" has already increased 50% of the way from its baseline to empirical values; in contrast, "morning B" has reached only around 2%. The large increases for "morning B" occur between $\tau = 300$ and $\tau = 540$, where its lower bound runtime increases around 90% of its baseline-empirical range.

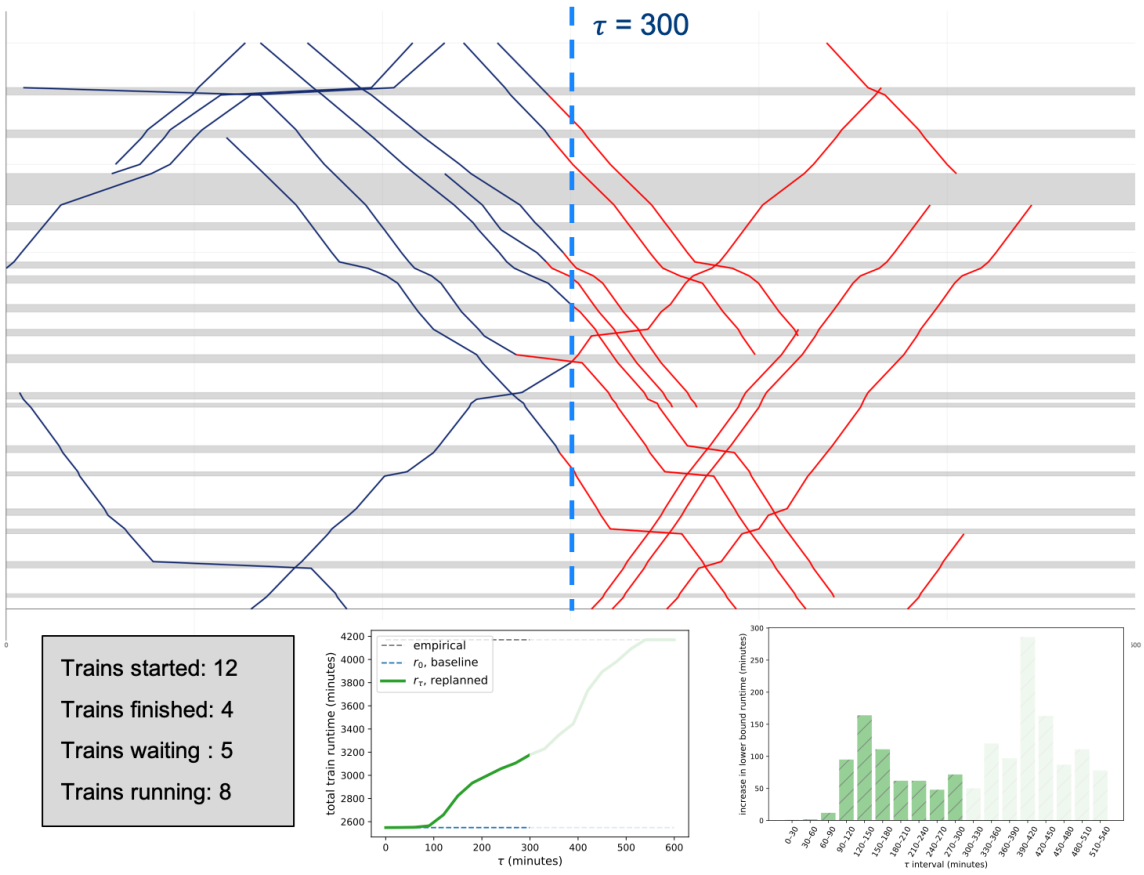


Figure IV.10: Depiction of replanning process at $\tau = 300$: stringline diagram of network state up to τ in blue and replanned future in red, replanned runtime plot (lower middle), and increase in lower bound per interval (lower right).

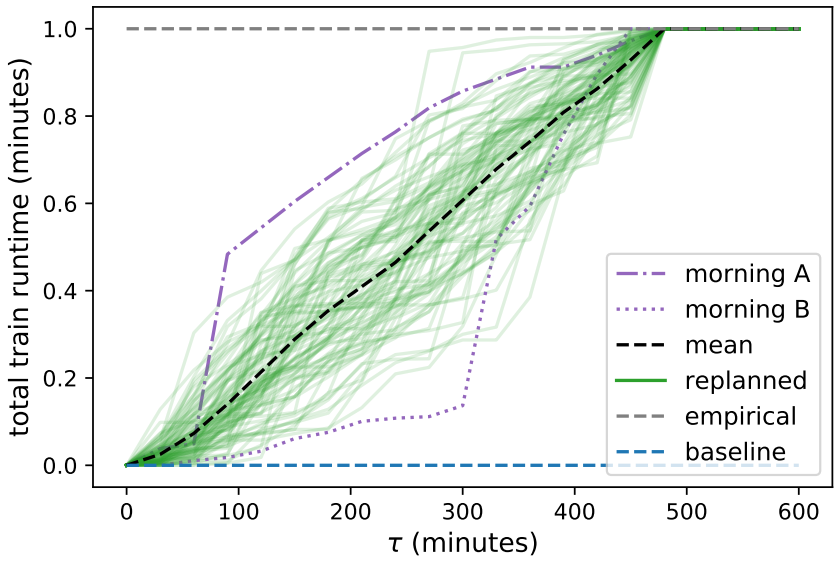


Figure IV.11: Accumulation of additional runtime due to replanning, shown for 90 windows of data, each of length 9 hours and overlapping by 6 hours. Each runtime value was min-max (baseline-empirical) normalized to [0, 1] for consistency. Different temporal patterns in the accumulation of delay can be seen in the green curves and the mean is shown by the black dashed line. Two dispatch windows, "morning A" and "morning B" are highlighted to demonstrate the vastly different patterns that occur.

IV.9 Problem 2: results

The analysis question at hand in problem 2 is: which alterations could have been made to the current network state in order to reduce the lower bound runtime under future replanning? Given that an amount of time has elapsed and empirical decisions have been made up to time τ that increase the lower bound runtime by $\Delta = r_\tau - r_0$, what could have been done differently that would have reduced Δ? Figure IV.12 shows, graphically, the effect of reducing runtime under replanning from r_τ to a lower value, denoted r', achieved by altering the network state at τ while replanning into the future. We first address this question at a specific value of τ and then show results for other values of τ in the same dispatch window.

The alteration of empirical decisions to reduce the overall runtime value from the replanned dispatch lower bound presents two bounding cases: 1) if no reduction in the lower bound runtime, r_τ, is desired, then no alteration of the empirical decisions is required; 2) if a reduction in the objective value back to its baseline optimal value, r_0, is desired, then the empirical data must be changed all the way back to the baseline dispatch plan (assuming the baseline plan was uniquely optimal).

For reductions in the overall runtime value between these two cases, we minimize the amount of *alteration* of the empirical data that is required. As a reminder, an "alteration" is defined in equation (IV.9) of the formulation of problem 2 as a change in a train's segment runtime, compared to its empirical value. Changing

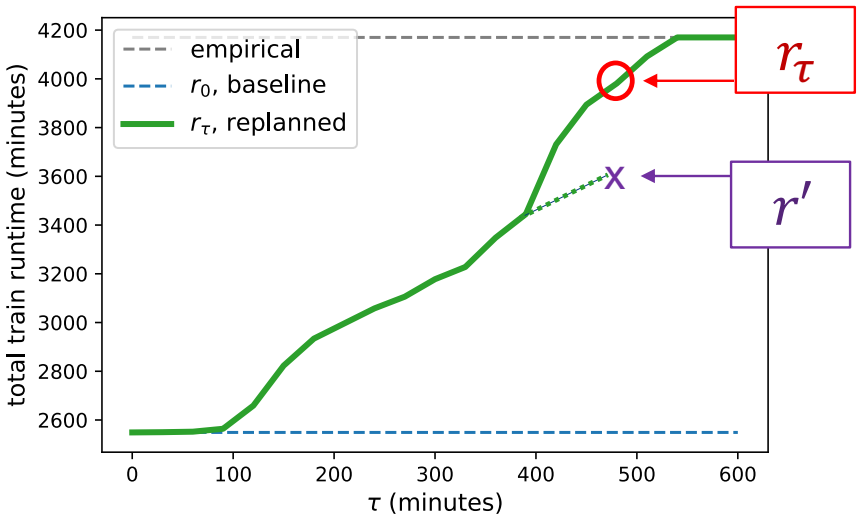


Figure IV.12: Representation of reducing lower bound runtime under replanning from its initial value of r_τ to a lower value r'.

a segment runtime naturally shifts the timing points for all successive track segments by the same amount. The $\mathcal{L}_1$ norm on alterations in equation (IV.9) promotes sparsity in the alterations that are found in the solution. The amount of *empirical alteration* is thus defined as the sum of these alterations and measured in minutes.

We first analyze the same 9-hour dispatch time period discussed in Section IV.8. At time $\tau = 300$, we solve the minimum empirical alteration problem for runtime reduction values of 0% to 100% in increments of 10%. Figure IV.13 shows the amount of alteration required (red curve) to reduce the overall runtime from its replanned lower bound value of 3178 minutes (green dashed line) to the baseline optimal value of 2549 minutes (blue dashed line). That is, for a reduced runtime of 2990 minutes (a 20% reduction), an alteration to empirical segment runtimes up to $\tau = 300$ of 45 minutes would be required.

The low slope of this curve at higher values of runtime (lower values of reduction) indicates that alterations to empirical data are yielding large effects on overall runtime. At the point where runtime is reduced to 2990 minutes, this reduction of 188 minutes was achieved by an alteration of 45 minutes, less than 25% of the magnitude. In this regime where the amount of empirical alteration produces a larger effect on the overall runtime, it must necessarily be affecting the secondary delay. Presuming that, in the 20% reduction case, the 45 minutes of alteration were all used to directly decrease (and not increase) runtime in the empirical trajectories, the alterations produced an additional 135-minute reduction in other runtimes that was the result of an improved ability to replan.

In Figure IV.13, a 1:1 line for empirical alteration (y-axis) to runtime reduction (x-axis) is also shown. A set of empirical alterations which produce a decrease in overall runtime only as large as the alterations

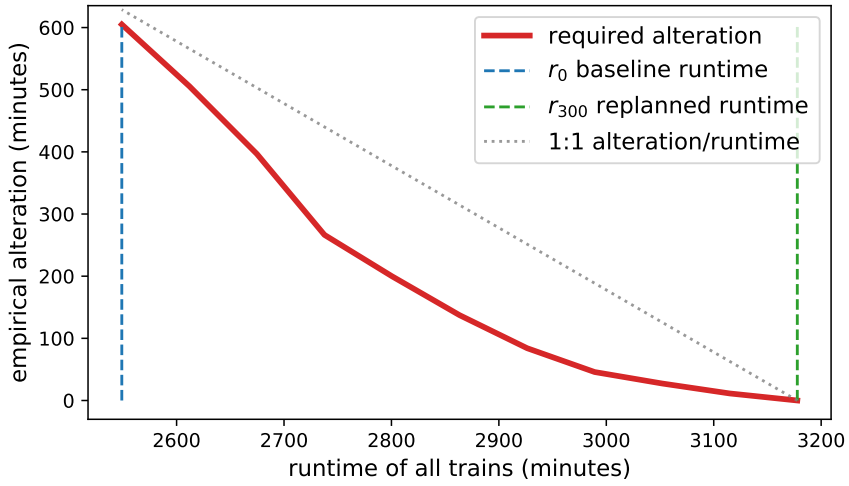


Figure IV.13: Required amount of alteration to $\tau = 300$ empirical data to reduce replanned runtime (approximately 3150 minutes, green dashed line) back towards its optimal value (approximately 2500 minutes, blue dashed line).

themselves would produce a slope on this plot the same as this 1:1 line. Therefore, we can interpret slopes of the alteration curve lower than this line as more efficient, producing magnifying effects on runtime reduction, and slopes greater than this line as producing inefficient effects on runtime reduction.

An example of one of these magnifying changes is shown in Figure IV.14. This stringline diagram shows two sets of trajectories: red trajectories are the replanned dispatch at $\tau = 300$, which accounts for empirical decisions made up to this time (marked by the blue dashed line); green trajectories are the replanned dispatch with changes made to empirical data which decrease the overall runtime. In general, we should see green trajectories complete before red ones, but in some cases tradeoffs can be made where some trains experience increased runtime but the total decreases. Two purple boxes highlight changes to the empirical trajectories of one train that reduced its runtime before $\tau = 300$ and allowed it to make a meet event with an opposing train at a siding further away, instead of waiting a long time at a closer siding. This change that was allowed is highlighted by the purple circle and arrow. Notice the downstream location where the meet between these two trains occurs, instead. This train's runtime is allowed to decrease over 40 minutes as a result.

The amount of empirical alteration, as described earlier, is measured by the magnitude of the $\mathcal{L}_1$ norm of changes made to track segment runtimes. This is the quantity that is minimized in the formulation for problem 2; the $\mathcal{L}_1$ norm also promotes sparsity in the alterations. The alterations highlighted in problem 2 are perhaps most useful if they are small, unique decisions that could have significantly changed dispatching. For example, a sub-optimal meet location which introduced unnecessary delay is a decision that could be easily investigated and perhaps corrected. From the same test that was described above, which generated the

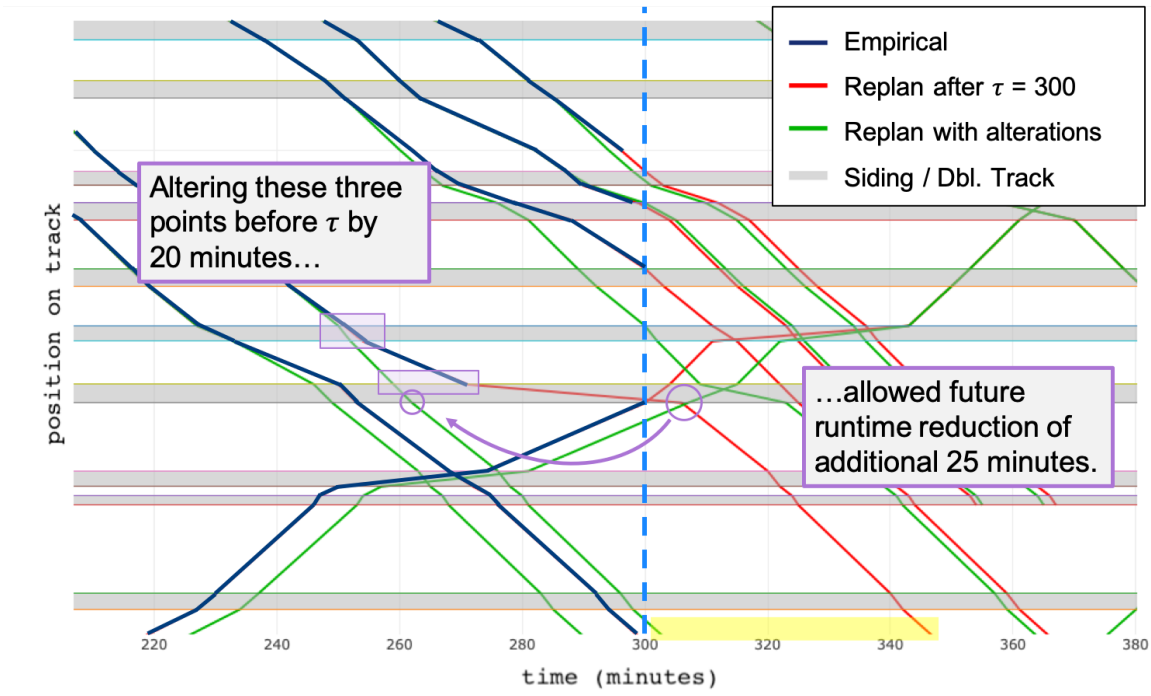


Figure IV.14: Example stringline diagram for reduction in replanning lower bound by 20%, compared to baseline plan, using empirical alteration. Lower bound replanning at $\tau = 300$ (blue dashed line) is shown by the red trajectories. The reduced lower bound with empirical alteration is shown in green. The two purple boxes highlight areas in which alterations to the empirical data were applied, which allowed a meet event to occur earlier at a downstream siding, removing a large delay shown by the purple circle and arrow. This train's runtime was reduced over 40 minutes as a consequence.

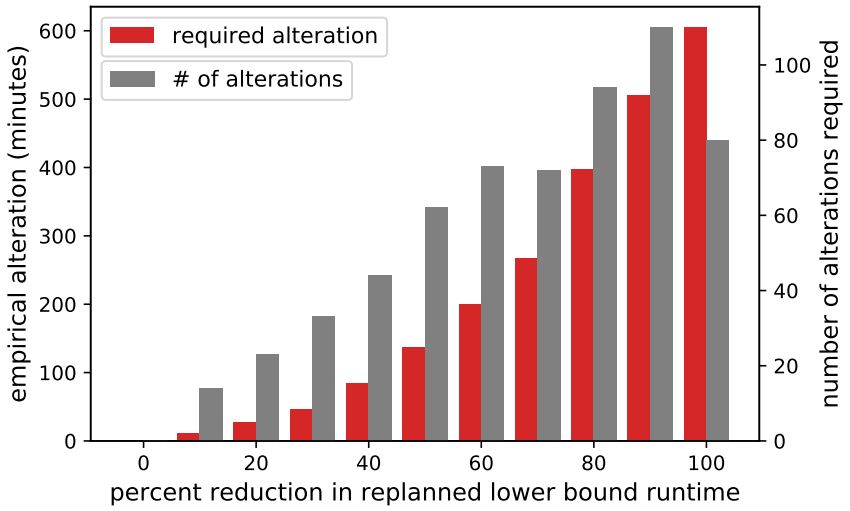


Figure IV.15: Comparison of the number of alterations and the magnitude of alterations required to reduce runtime at $\tau = 300$ from its replanned lower bound.

required alteration curve in Figure IV.13, we extract not only the magnitude of alteration, but the number of track segment runtimes that were changed. For each percentage reduction in lower bound runtime (0% to 100%, increments of 10%), Figure IV.15 shows the number of distinct alterations (grey hatched bars) alongside the magnitude of alteration (red bars). We see that a runtime reduction of 20% can be achieved by changing 23 segment runtimes in the empirical data. While this is a modest number, the trend does not exhibit the same degree of positive effects that are seen with the magnitude of alteration; the number of alterations is more linear with respect to runtime reduction than the magnitude of alteration.

Thus far, we have presented correction results only for $\tau = 300$ minutes. Figure IV.16 shows the same curves for empirical alteration required to produce runtime reduction at values of $\tau = 120$, 180, 240, 300, 360, 420. A similar trend is observed for other values of τ on the same window of dispatch data. Large initial gains in overall runtime reduction (e.g., 10%, 20%) are possible with a very small degree of alteration to empirical data, after which returns diminish. The baseline lower bound is denoted by the blue dashed line. The runtime reduction for $\tau = 420$ is slightly more aggressive for small amounts of empirical alteration. This could be due to a larger set of decisions that can be altered or greater secondary effects of those decisions that have propagated to other trains.

Figure IV.17 shows the magnitude of empirical alteration and the number of unique alterations for percentage reduction values in this same dispatch window. Interestingly, the 20% runtime reduction for $\tau = 420$ requires fewer alterations than the equivalent reduction for $\tau = 240, 300, 360$. This supports the idea that at $\tau = 420$ there are a larger set of impactful decisions that could be mitigated, or dispatch decisions have

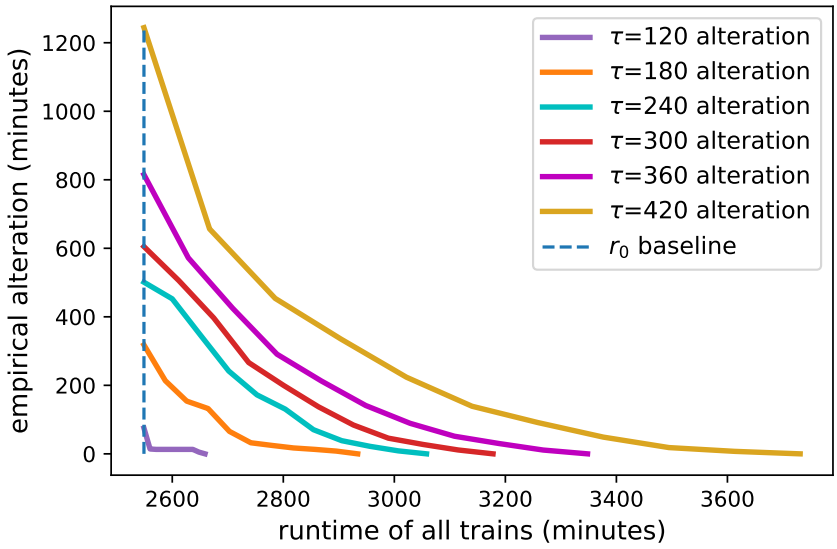


Figure IV.16: Required alteration to empirical data at various values of τ to shift replanned runtime (not shown) toward the optimal plan value of approximately 2500 (blue dashed line).

caused greater secondary effects that can be reduced by altering the initial decision.

IV.10 Problem 3: results

Problem 3 addresses the effects that a single train in the dispatch plan can have on other trains in the same plan. Specifically, what is the cost to the dispatch plan of fixing the trajectory of a single train to its empirical value. We first discuss results on a single window of dispatching and analyze a specific train in detail within this time period. Then we look at broader trends exhibited by trains on this network segment within a month of dispatching.

As discussed in Section IV.5, we define primary added runtime for a given fixed train w to be the difference between its empirical runtime, γ'_w and its baseline dispatch value, γ. Secondary added runtime is computed as the increase in runtime of all trains, minus the primary added runtime of train w: $(r_w - r_0) - (\gamma'_w - \gamma_w$. It is possible for either the primary or secondary values to be negative, but not both; the sum of primary and secondary added runtime must be positive because the baseline dispatch plan was globally optimal and no possible trajectory for the fixed train may decrease the overall runtime value. In the case that a train runs faster than its optimal trajectory, it can do so only at the expense of increasing runtime for other trains by at least as much. In the case that secondary added runtime is negative, it is possible only because the fixed train experienced increased runtime.

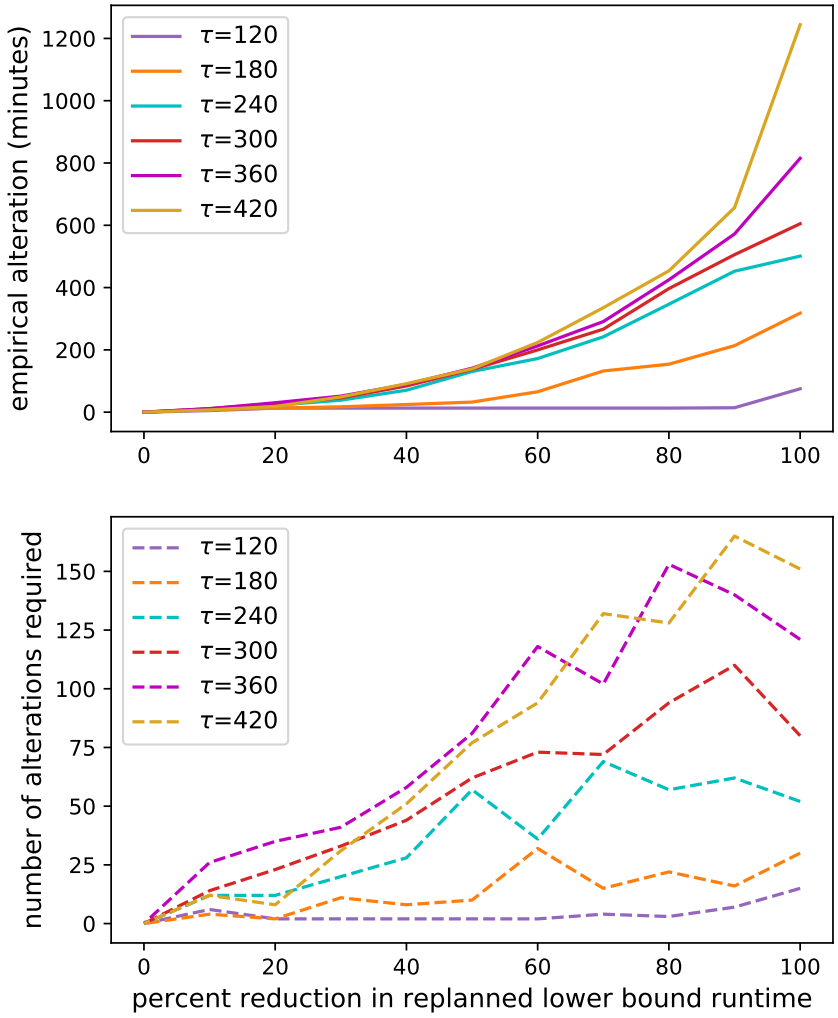
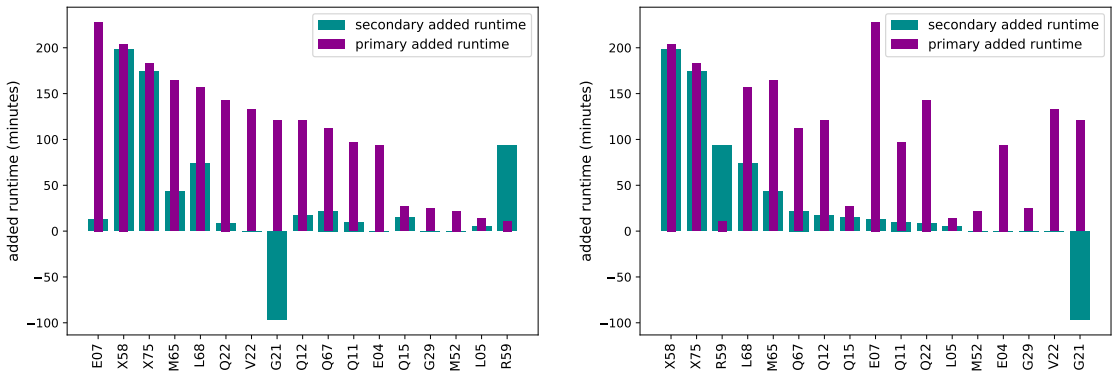


Figure IV.17: Comparison of the number of alterations (bottom) and the magnitude of alteration (top) required to reduce runtime for this dispatch window by a given percentage. Values of τ from 120 to 420 minutes are shown.



(a) (b)

Figure IV.18: Comparison of primary added runtime incurred by trains and their minimum induced secondary added runtime on the optimal trajectories of other trains. Trains in the same window of data are sorted by primary added runtime (a), and by secondary added runtime (b).

IV.10.1 Analysis on a single dispatch period

We run the fixed-train dispatch for each train in the 9-hour dispatch window during the week of January 4, 2016, as described in Section IV.7. Figure IV.18 shows the primary and secondary added runtime caused by each train when it was fixed in the dispatch. Figure IV.18a sorts the set of trains by primary added runtime. Two of the top three trains in terms of their own added runtime were the top two contributors to the added runtime of others, when fixed: X58 and X75. However, the train experiencing the greatest primary added runtime caused almost zero secondary effect. At the other end of the spectrum, train R59 experienced very little primary added runtime, but the secondary effect was much larger. This train will be analyzed in more detail, later.

In Figure IV.18b, the same set of trains are sorted according to secondary added runtime. At the opposite end of this spectrum, fixing train G28105 caused a substantial decrease in runtime for other trains, but at the cost of its own runtime. Note that the net effect is still added runtime, but it shows that running the schedule around the departure time of train G28105 slowed down the rest of the trains, significantly. This effect indicates that placing G28105 on the network at a less sensitive moment could likely have improved overall performance.

Let us now analyze the case of train R59 in more detail. This train experienced a very small amount of added runtime with respect to the optimal dispatch plan, but this small primary effect caused an outsize secondary effect, over 5x larger. Consider the partial stringline diagram for train R59 in Figure IV.19, which shows a magnified area of the diagram in the vicinity of the train. Train R59 completes only 5 network segments and incurs 11 minutes of added runtime, shown by the red trajectory, relative to its optimal trajectory, which is shown in blue to its left. This shift in the trajectory causes the two trains following in the same direc-

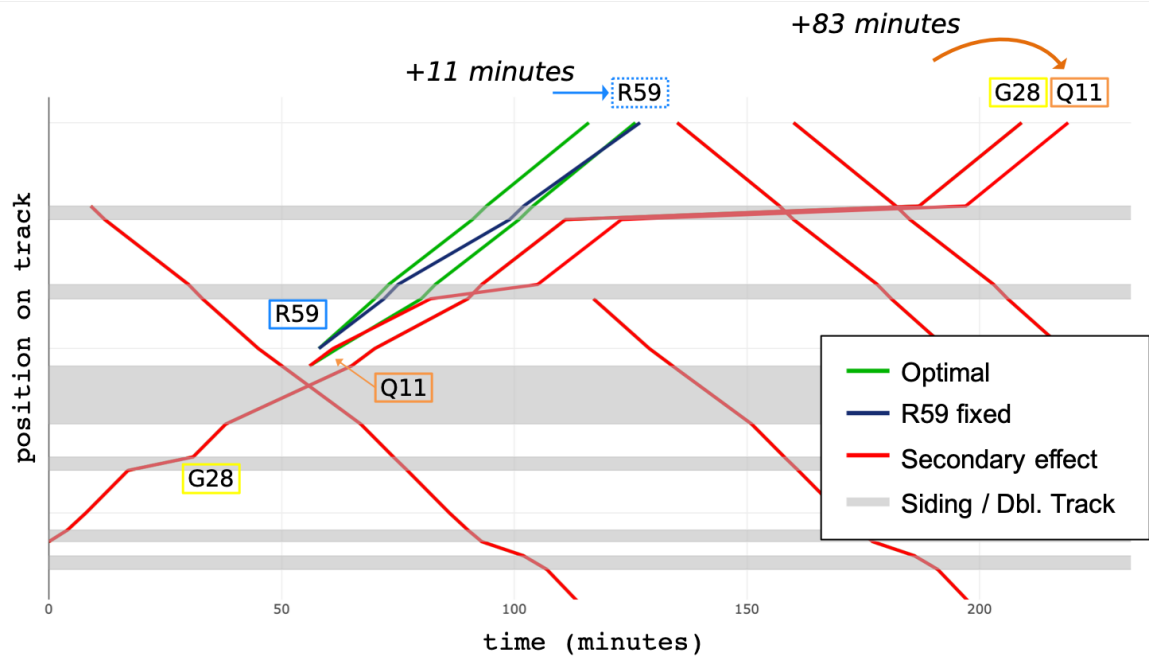
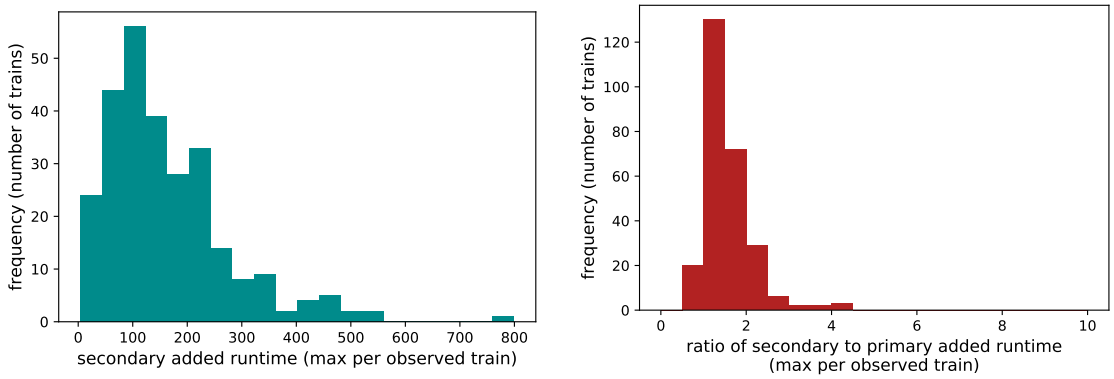


Figure IV.19: Portion of a stringline diagram with one train, R59, shown in red, fixed to its empirical data. The optimal plan is shown in blue, behind the replanned green trajectories that consider the fixing of train R59. An added 11 minutes of runtime for R59 resulted in a delay of at least 83 minutes for successive train Q11.

tion, Q11 and G28, to run slower in order to maintain the required headway. Q11 was supposed to complete its run immediately after R59, before a train in the opposite direction departed. However, the delay required Q11 to wait out on the final siding for two trains in the opposite direction to clear, before it could complete its route. The secondary effect was at least 83 minutes for Q11. The second following train, G28, was also affected with an added runtime of approximately 10 minutes, but its optimal trajectory happens to be hidden behind the secondary effect trajectory for the train Q11.

IV.10.2 Results across multiple periods

The findings presented thus far concern only a single dispatch window with 17 trains, but can also be analyzed for a longer range of data. We now evaluate a shifting dispatch window of 9 hours over a month of data, January 2016, with overlap of 6 hours between windows. Since a train can be observed in multiple dispatch windows due to overlap, the primary and secondary added runtime effects are selected for each train from the dispatch window where secondary effect reaches its observed maximum absolute value. The distribution of secondary added runtime effects for all observed trains observed in the month is shown in Figure IV.20a, clipped to the upper limit of 800 minutes. A large number of trains incur over 3 hours (180 minutes) of secondary added runtime and a few, approximately 20, incur over 5 hours. These trains would merit further



(a) (b)

Figure IV.20: Distribution of the secondary added runtime effects of trains (a) and the distribution of ratios of secondary/primary added runtime effects for each train (b).

investigation into their precise configuration.

Secondary effects are highly dependent on the primary effect that induced them. We therefore consider the ratio of secondary to primary added runtime effects in Figure IV.20b. The vast majority of trains have less than a 2.0 ratio, meaning that secondary effects were less than twice the value of primary effects, but approximately 40 exceeded a 2.0 ratio value and a few were past a 3.0 ratio value, again indicating high relative impact of these trains on the schedule.

Finally, we observe in Figure IV.21 these primary-secondary pair values on a scatter plot. Primary added runtime is on the x-axis and secondary on the y-axis. A linear trendline for the dataset is shown by the dotted black line, which has a slope of 1.4 minutes/minute. This means that on average, a train adds 1.4 minutes of secondary runtime to other trains for every minute of its own runtime in excess of its optimal trajectory.

IV.11 Conclusion

In this Chapter, we present a methodology by which to analyze empirical dispatch performance with regard to its optimal dispatch plan. This methodology is applied to answer three principle questions: How did a current network state contribute to a deterioration in the optimal dispatch plan? Which specific changes could have hypothetically been made to the network state that would have reduced the cost of replanned dispatch? Which trains' performance caused an effect on others in the schedule in terms of inducing additional runtime?

These three questions are addressed using the proposed *dispatch analysis problem*, which follows from a common form of optimization-based dispatching. The general form of the dispatch analysis problem accommodates each of these three questions by changing only the objective function and a few key constraints.

We apply the methodology to the questions identified and show that in identifying the deterioration of the

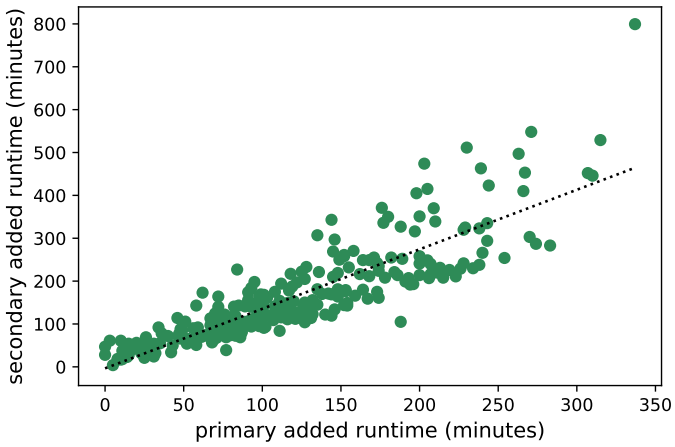


Figure IV.21: Scatter plot of primary (x-axis) versus secondary (y-axis) added runtime for a month of trains. Each point is a train's primary/secondary values in the dispatch window for which it exhibits the largest secondary added runtime value. The dotted black line is a linear trendline. Its slope is 1.4 minutes/minute, indicating 1.4 minutes of secondary added runtime for every minute of primary added runtime by each train.

optimal dispatch plan, it can construct a timeline of how the lower bound dispatch performance under optimal replanning increases. Sharp increases in this lower bound indicate decisions that were costly to the dispatch plan immediately and into the future. Each window of dispatching data exhibits a unique pattern around which this deterioration manifests. The application for alteration of the empirical data to reduce sub-optimality of the replanned dispatch demonstrates that small changes to empirical data can have a magnifying effect on the reduction of total runtime. In one particular case, a magnifying effect of 4:1 was observed, meaning that, effectively, each minute saved in the past would have saved 4 minutes in the overall plan. Finally, the analysis of specific trains in a dispatch window revealed that trains have highly non-uniform effects on the schedule. A small number of trains have secondary effects on the schedule that far exceed the effect of their own deviation from the optimal schedule. Taking these extreme cases, in and of themselves, could reveal useful improvements to relevant scheduling and dispatching practices.

Overall, we believe this methodology, being a natural extension of optimization-based tools that exist in practice, can serve to become a powerful analysis and examination engine for empirical dispatching practices. In future work, adding fidelity to the dispatching model could help create more realistic train trajectories via optimal dispatch. Also, extending the application to other territories and analyzing aggregate effects could reveal larger trends or overall dispatching performance.

CHAPTER V

Estimating freight train arrival times

V.1 Introduction

V.1.1 Motivation

The rail network in the United States has significant infrastructure capacity limitations that cause congestion of the rail traffic. Few rail corridors contain exclusively double (or more) track that allows simultaneous bi-directional traffic (Murali et al., 2010). In comparison, the double and triple track railroads in Europe provide for double the train density of US rail networks (Wyman, 2016). Many US corridors contain a single track with short sections of double track known as *sidings*, where trains may meet or pass each other. These *movements* (i.e., meets, passes) are implemented through the railroad signaling system, but are directed by human dispatchers. Dispatchers are experienced with working on specific track corridors, but movements on sidings require planning and precise timing in order to achieve efficient operations (Vromans et al., 2006; Kecman and Goverde, 2013). Freight volume is expected to increase in the US, so either infrastructure capacity must be increased or operational improvements must be made to increase capacity (Cambridge Systematics, 2007; Weatherford et al., 2008; Association of American Railroads, 2013).

In addition to the track infrastructure constraints, there are numerous other factors that can contribute to variability of the runtime on a track segment. Traffic heterogeneity and the train priority differences directly influence both the runtime of trains and also the variability in the runtime (Dingler et al., 2009, 2010). Physical characteristics of trains such as the length, tonnage, and power further influence the runtime due to track grade, track curvature, and siding lengths (Dingler et al., 2009). The ability of a train to complete a trip and exit the *line of road* (i.e., the track segments connecting distant terminals) is also influenced by the degree of congestion in the arrival terminal. This is compounded by the possible actions required for the train in the terminal, such as refueling, inspection, switching of cars, or crew change (Dingler et al., 2009; Higgins et al., 1995). Railroad operating strategies such as dynamically scheduled trains and maximizing train length are particularly vulnerable to delay (Lu et al., 2004; Mu and Dessouky, 2011).

In the presence of runtime variability, ETAs are necessary in order to improve real-time decision making and the efficiency of the network (Hertenstein and Kaplan, 1991; Hallowell and Harker, 1998). For example, future train schedules can be continually updated to provide new train plans to allow traffic to flow smoothly between terminals on the network Kraay and Harker (1995). Although there are many techniques available to derive optimal schedules (see Goverde (2005) for a thorough review), the schedule may be very sensitive to

delays when the network is near capacity. High capacity utilization leads to more complex dispatching where small delays are created, leading to larger deviations from the train plan (Khoshniyat and Peterson, 2017); this is referred to as *knock-on delay* (Vromans et al., 2006; Murali et al., 2010; Goverde and Meng, 2011).

Highly variable runtimes increase operational uncertainty for the railroad and for other transportation systems that depend on them. On the rail network, propagation of delay to other trains is significant (D'Ariano and Pranzo, 2009), and there are large direct costs incurred due to additional operating time alone (Lovett et al., 2015). Delays on the rail network can also influence non-rail transportation services. For example, surface street traffic and emergency vehicles, which conflict with rail freight traffic at grade crossings (Estes and Rilett, 2000), can be significantly delayed if a grade crossing is occupied by a train for an extended period of time. If accurate, real-time ETAs are made available, revisions to the operating plan can be implemented, and surface street transportation services can be re-routed.

V.1.2 Problem statement and related work

The main focus of the present chapter is the prediction problem for ETAs on US freight railroads using real-time data. The estimation problem requires new ETAs to be produced as time elapses and the train progresses down the line of road. Each time the train reaches one of a number of fixed locations on the track, data is collected and a new estimated travel time to the destination is produced.

To produce the ETA estimate, a variety of routinely collected and maintained data sources available to freight railroads are used. This includes track geometry data (containing grade and curvature information, single and multi-track territory, length of sidings, etc.), historical runtime profiles of all trains, properties of all trains (such as length and tonnage), and crew records.

V.1.3 Outline and contributions

The main contribution of this work is to show how to pose the ETA prediction problem at grade crossings on a rail network as a machine learning regression problem and to provide results indicative of predictive performance across a range of time horizons and various machine learning algorithms. ETA updates occur at fixed timing points on the network and can be generated corresponding to any other timing point, which can be grade crossings, or to major destinations on the network. We provide practical insights by highlighting the datasets available to perform the prediction and describing some of the feature engineering required when the feature vectors change in time and space. We present a set of data features and several machine learning regression algorithms used to achieve more accurate ETAs than common statistical methods yield. Finally, the resulting models are discussed in detail with respect to their performance.

Due to multiple approaches taken in this work, we first present a single origin-destination modeling

framework that is straightforward to understand, followed by a unified all-origin framework that can leverage larger training datasets and predict ETAs from multiple locations in a single model. Both of these models are applied first to ETA predictions made to train destinations at yards and terminals, so that we may benchmark machine learning techniques against existing algorithms used by the railroads and compare the performance of various machine learning algorithms.

The remainder of the chapter is organized as follows. Section V.2 presents the framework used to process and operate on the various data sources and the preliminaries for the machine learning regression. Section V.3 discusses the datasets and the work that is necessary to process the data for use in the machine learning framework. Section V.4 details the model experiments that are conducted as well as the means by which to evaluate them. Section V.5 describes the process for tuning and evaluating one example of the origin-destination models, which is then extended to models across the full testing route; results are given for models using select feature combinations. Section V.6 formulates and tests a unified framework to combine models and leverage larger quantities of data, and also compares performance of a variety of algorithms. We provide a summary and discuss future lines of research in Section V.7.

V.2 Framework and Problem Formulation

This section briefly describes the machine learning framework used for ETA estimation. It reviews general machine learning terminology and parameters specific to the algorithms, both used later during analysis and discussion.

V.2.1 Assumptions

We briefly summarize key assumptions that are made in the ETA estimation methodology and comment on their importance for consideration during potential implementation.

This work assumes that train arrival at a location is determined by the time at which it records its on-station time, or OS-point time. For the endpoints of network segments where yards are often located, the last OS-point is assumed to be the arrival point. Therefore, this point is assumed to determine arrival time relevant for crew on-duty time. This is a simplifying assumption due to data availability; an implemented system may have additional data resources to resolve this point.

Additionally, we assume the capability of an implemented system to extract and process data streams in real time. Location and status of trains across the rail network is needed in order to construct data features for inference of the machine learning models. Track slow orders, maintenance incidents, and other relevant events are needed to filter out trains for which ETAs will be invalid, or to provide alternate ETAs.

V.2.2 ETA Machine learning framework

The problem of predicting an estimated time of arrival for a train from an origin point to a destination point on the rail network is posed as a supervised machine learning regression problem. The goal of the regression problem is to predict the true runtime $y(i) \in \mathbb{R}$ of a train i given the properties of train i, the network, and other traffic on the network, which are contained in the feature vector $x(i) \in \mathbb{R}^n$. Given a dataset of m trains with true runtimes $Y = [y(1), y(2), y(3), \cdots, y(m)]^T \in \mathbb{R}^m$ and corresponding feature vectors $X = [x(1), x(2), x(3), \cdots, x(m)]$, where $X \in \mathbb{R}^{n \times m}$, the machine learning regression problem is to find a mapping $f_w \colon \mathbb{R}^n \to \mathbb{R}$ parameterized by w such that $f_w(x(i))$ is an accurate predictor of $y(i)$. In general, supervised machine learning regression uses a set of *training data* $\{X_{tr}, Y_{tr}\}$ (where the subscript *tr* is used to indicate the training data) to learn the function f_w, by minimizing a prediction error measure between $f_w(X_{tr})$[1] and Y_{tr} over the m records in the training data.

The machine learning model f_w must generalize (i.e., make good predictions on data that has not been used to train the model), in order to maintain high accuracy on new data and to avoid *overfitting* the training data. To test the degree of generalization, the accuracy of the prediction is assessed on hold out *test data* $\{X_{te}, Y_{te}\}$, which is not used to train the model.

V.2.3 Generating ETAs on a freight rail network

The central difficulty of posing the train ETA prediction problem into the framework above stems from the fact that many of the features used for prediction change in time and in space as the train moves towards the destination. For example, the amount of traffic on the line of road will change as trains enter and leave the line of road. The number of available sidings on a route in single track territory also changes across the route and as other trains occupy or vacate sidings. If a single model is used for all origin-destination predictions in the network, it may be difficult to predict area-specific delays (e.g., due to local dispatching decisions and route characteristics) that may not occur throughout the network; results pertaining to origin-destination specificity are discussed in Section V.5.2. Moreover, because some features change over time (as described above), while others may not (e.g., train priority), construction of an unbiased training dataset is a nontrivial engineering task. For example, one cannot simply create a new training data point each time a single property of a train changes (e.g., corresponding to a new feature vector) without biasing the training data, since the feature vector still corresponds to a single train trip.

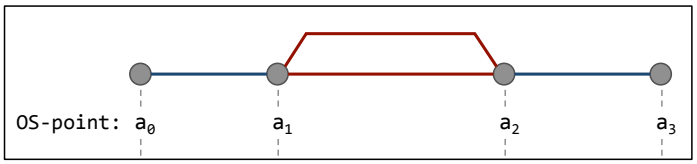
To address these difficulties, we propose to build a distinct regression model for each origin-destination pair for which predictions are required, where the ETA corresponds to the estimated time of arrival at the next destination (i.e., major terminal) for trains passing the corresponding origin point on the network. The

[1] With a slight abuse of notation, we overload the function f_w to also operate on the entire dataset X_{tr}.

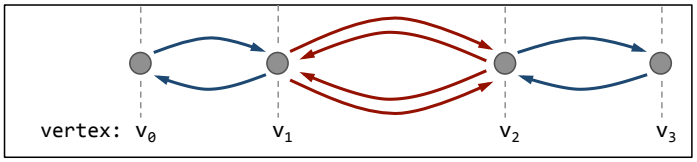
resulting models are all of the same form and differ only in feature weights and hyper-parameters. Because the models are independent, each model can be trained using all trips that pass between the corresponding origin-destination pair by constructing features according to the state of the train and network at the time the train reaches the origin point. Localized and geography-specific performance effects may be captured in the individual models without explicitly constructing them in the feature vector. For example, longer travel times will be observed for heavy or under-powered trains in areas of high track grade. Feature construction can also vary between models (e.g., in dimensionality) since each uses a custom dataset built for the origin-destination pair.

The primary disadvantage of building a model for each origin-destination pair in the network is the number of models required. In a rail network with k nodes, at minimum k^2 origin-destination predictors could be required (possibly more if multiple paths exist between each origin-destination pair). In contrast to road networks where the number of nodes and the number of viable paths between any two node pairs may be large, rail networks have fewer nodes and less path redundancy. In practice, few locations are relevant destination points from a given origin because a single route (excluding small deviations for sidings) is typically used to connect two points on the network. The freight rail network in the United States is sparsely connected in most regions, particularly with regard to high volume routes; isolated corridors connect major terminal points on the network where most crew changes and switching work occur. Therefore, the number of origin-destination paths for which predictions are required is tractable. In the area of study in this work, there exist 35 points that can serve as origin points. This results in 35 ETA updates for a train traversing the corridor. There are less than four practical destination points from each origin point, which results in at most 140 predictors as opposed to $35^2 = 1{,}225$. For a Class-I rail network in the US, we estimate a total of 10,000 models are necessary for all ETA predictions.

In order to map spatiotemporal train data to the network topology, infrastructure data can be reconstructed into a directed graph format, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of directed edges (Kecman and Goverde, 2015). Vertices are points where the track merges and splits (e.g., endpoints of sidings). Data on passing trains is recorded at *OS-points*, which are fixed locations $v \subset \mathcal{V}$; OS-points serve as the origin points in the origin-destination models. Directed edges represent track segments across which trains travel between OS-points and a direction of travel. This alignment between track infrastructure and the constructed graph is shown in Figure V.1. In the figure, OS-points are denoted a_0, a_1, a_2, and a_3 with corresponding graph vertices v_0, v_1, v_2, and v_3. All tracks are delineated between these OS-points with multiple tracks such as the main line and siding between a_1 and a_2 remaining distinct. Pairs of directed edges representing each delineated track segment allow different runtimes and feature values in each direction of travel, which is necessary when properties such as grade are considered (Bonsra and Harbolovic, 2012). This results in two



(a) Track infrastructure representation assembled from GIS data.



(b) Vertex-edge graph constructed from track infrastructure data.

Figure V.1: Graph vertices align with OS-points and each track segment between them is represented by a graph edge in each direction. Double track areas and sidings, therefore, are represented by four graph edges.

directed graph edges for a single track and four or more edges for an area of multiple tracks such as the siding between vertices v_1 and v_2. In this formulation, all trains can be routed on the graph across their unique path considering track usage (e.g., siding track versus main line track). Data can be gathered on the behavior of trains for each directed edge with respect to speed, grade, train occupancy, and other location/direction specific attributes. Also, features that consider estimates of the positions of multiple trains and track topology can be mined from this data.

V.2.4 Regression via support vector machines

The regression problem of predicting an ETA from a vector of features is proposed to be solved via *support vector regression* (SVR), introduced in Drucker et al. (1997). Support vector regression is a popular machine learning algorithm grounded in statistical learning theory and for which training is efficient due to the convexity of the training problem. The optimal model parameters in linear SVR are straightforward to interpret and are unique (Burges and Crisp, 2000), which can be invaluable in the application of the algorithm. Additionally, the SVR formulation provides for extension to nonlinear regression via kernel functions. The intent of this work is not to demonstrate superiority of SVR to other algorithms, but to apply a well-studied algorithm to the data-driven ETA prediction. The precise algorithm that should be implemented in a live production system would depend on performance and additional practical factors, such as computation time, memory requirements, and more. Other applicable algorithms include linear ridge regression (Hoerl and Kennard, 1970), elastic net regression (Zou and Hastie, 2005), kernel ridge regression (Saunders et al., 1998), random forests (Kecman and Goverde, 2015), and neural networks (Marković et al., 2015), to name a few.

The training step in a generalized regression problem may be written as:

$$\min_{w} \quad L(f_w(X) - Y) + \|w\|, \tag{V.1}$$

where L is a loss function measuring the quality of the predicted output, $f_w(X)$, relative to the true output, Y, and the feature weights w are penalized via a norm to avoid overfitting the training data. This characteristic is informally referred to as model *flatness* (Basak et al., 2007).

SVR is a special case of (V.1) and uses a two-norm on w and in the simple case assumes an affine predictor of the form $f_w(x) = w^T x + b$, where $w \in \mathbb{R}^n$ and the offset $b \in \mathbb{R}$. SVR uses an *ε-insensitive loss function* $|\cdot|_\varepsilon$, which penalizes prediction residuals $r = y - f_w(x)$ larger than a threshold defined by ε. The loss function is constructed as (Cortes and Vapnik, 1995):

$$|r|_\varepsilon = \begin{cases} 0 & \text{if } |r| \leq \varepsilon \\ |r| - \varepsilon & \text{otherwise.} \end{cases} \tag{V.2}$$

The ε-insensitive loss function quantifies the distance between a prediction and the band created by $y \pm \varepsilon$. For a vector of residuals, the sum of the element ε-insensitive losses are used as the loss function.

The training step in SVR can be reformulated as computing the weights w and offset b by solving the following convex optimization problem:

$$\begin{aligned} \underset{w,b,\xi,\xi^*}{\text{minimize}} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}(\xi(i) + \xi^*(i)) \\ \text{subject to} \quad & y(i) - w^T x(i) - b \leq \varepsilon + \xi(i), \quad \forall i \\ & w^T x(i) + b - y(i) \leq \varepsilon + \xi^*(i), \quad \forall i \\ & \xi(i),\, \xi^*(i) \geq 0, \quad \forall i \end{aligned} \tag{V.3}$$

where $\xi, \xi^* \in \mathbb{R}^m$ are variables introduced to rewrite the ε-insensitive loss (V.2) as linear inequalities. The total ε-insensitive loss (i.e., accuracy of model fit) is balanced against model flatness by a scalar factor C.

The optimization problem (V.3) can be solved via the dual problem, yielding the optimal dual variables $\alpha, \alpha^* \in \mathbb{R}^m$. These dual variables are related to the feature weights such that $w = \sum_{i=1}^{m}(\alpha(i) - \alpha^*(i))x(i)$. This results in a predictor of the form:

$$f(x) = \sum_{i=1}^{m}(\alpha(i) - \alpha^*(i))x(i)^T x + b; \tag{V.4}$$

see Scholkopf and Smola (2001) for a comprehensive description.

When the ETAs in the training data are not linearly related to the features, an alternative strategy is to transform the training data into a much higher dimensional feature vector denoted by $\Phi(x)$, where $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^N$ with $N >> n$, which can then be used for regression. The new predictor becomes:

$$f(x) = \sum_{i=1}^{m} (\alpha(i) - \alpha^*(i)) \Phi\left(x(i)\right)^T \Phi(x) + b. \tag{V.5}$$

Interestingly, it is not necessary to explicitly define the mapping to the high dimensional space, since only the inner product $\Phi^T \Phi$ is needed in the regression function. The inner product can instead be defined through a kernel function $K(x(i), x(j)) = \Phi\left(x(i)\right)^T \Phi(x(j))$. The use of the kernel function directly in (V.5) is known as the *kernel trick* (Burges, 1998) in machine learning. In the present work, we adopt the *radial basis function* (RBF) kernel (Boser et al., 1992) of the form:

$$K(x(i), x(j)) = \exp\left(-\frac{1}{2\sigma^2} \left\| x(i) - x(j) \right\|_2^2\right), \tag{V.6}$$

where σ is a parameter controlling the decay rate of the kernel, effectively limiting the influence that any single observation may have on the trained model.

V.2.5 Random forest regression

Random forest regression is an ensemble algorithm that constructs a series of regression trees using randomly sampled subsets of the training data for each tree and a subset of available data features for splitting within trees (Breiman, 1984, 2001). Regression trees are constructed by splitting data samples at each node in the tree according to values of input features. Each node resulting from the split more effectively isolates data samples with similar output values. The best split is determined by a minimization of resulting prediction error. Nodes are no longer split when the number of samples in the node falls below the minimum or the decrease in prediction error falls below a defined threshold. The predicted output value for each terminal node in the tree is calculated from the corresponding training samples that terminated in the node. The predictions made by individual trees are averaged to arrive at the ensemble prediction. Combining many weak learner regression trees in the random forest predictor has shown to be an effective methodology and helps avoid overfitting (Liaw and Wiener, 2002; Oruganti et al., 2016).

V.2.6 Deep feed-forward neural network model

A neural network consists of multiple neurons organized in layers, with individually weighted connections between neurons in adjacent layers. At minimum, a feed-forward neural network consists of three distinct

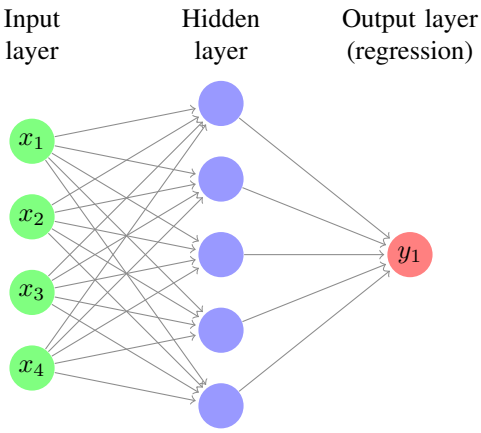


Figure V.2: Simple Feed-forward neural network with one hidden layer.

layers: the input layer, one hidden layer, and an output layer consisting of one node for regression, as shown in Figure V.2, or multiple nodes for classification. A deep feed-forward neural network has multiple hidden layers. The depth of the model is determined by the number of hidden layers. After being processed at the hidden layer nodes, their outputs are forwarded to the output layer which then makes a prediction, according to its activation function. This feed-forward process is characteristic of the neural network and is used for making predictions, given an input vector. The training phase of a neural network is comprised of selecting the optimal weights for each of the connections between the neurons. More specifically, given the input vector at the input layer, and the known output that actually occurred, the problem is defined as choosing the weights for the connections between neurons so as to minimize the error between the prediction and actual observation. Gradient-based optimization is used for training the neural network and choosing weights.

V.3 Feature construction and data cleaning steps

This section discusses the process of combining and mining datasets to be used in feature construction. The data used in this work is described first, before describing the features that are calculated from the datasets which are subsequently used to train the machine learning ETA algorithm. Due to the proprietary nature of the data, some values are reported in relative terms.

V.3.1 Description of raw data

This work uses a collection of datasets describing the rail network and operations from December 1, 2014 through January 31, 2017 inclusive. It consists of freight train movement, train car operations, crew, and locomotive data in the CSX Transportation network extracted from dispatching, operations, and signaling data.

The movement data consists of records generated at OS-points between terminals. The data includes the track on which the train was reported and the time at which the train triggered the OS-point. This dataset also contains information about track mileage covered, direction of travel, and the next destination at which the train is scheduled to stop. OS-points have spacing between 1 and 10 miles and typical temporal spacing between 1 and 20 minutes. Typical runtimes for the full route vary between 5 hours and the maximum crew time of 12 hours.

The train car operations data details the actions performed on the train once it enters a terminal from the line of road. This includes the switching operations (i.e., picking up and setting out rail cars) that are referred to as train *work*, inspections, refueling, and crew changes that are scheduled to occur and may incur delay in getting track space in the terminal. The planned work schedule, as well as adherence to the schedule, is reported in this data. Changes in physical train characteristics (e.g., total number of cars, length, tonnage) are inferred based on the work reported on the train.

Crew data contains information about the crew assigned to the train, the originating location, the time at which they were called on duty, and the time at which the crew must legally go off duty (i.e., 12 hours after going on duty). The time between a crew going on duty and the departure of the train is non-negligible and is referred to as *on duty time to departure* (ODTOD). Crew information is important because of maximum allowable on-duty time. This on-duty time requirement must always be satisfied, even at the large expense of stopping a train and transporting a replacement crew to finish the trip.

Locomotive assignment data indicates the equipment and total locomotive power available on each train, which can be important for predicting delays in regions with high grades.

This work also uses GIS data describing the physical infrastructure of the network, which includes individual tracks, switches, mileposts, and terminals. All locations referenced in the movement, work, crew, and locomotive data map to physical infrastructure locations, such as track mileposts and control points. Reconciling these GIS data sources is necessary to gather data on the number of tracks and siding locations and lengths on each route and build the network graph described in Section V.2.3. Figure V.3 depicts this data, along with the distinction between single track sections (shown by the thin blue lines) and multiple track sections or sidings (shown by the bold red lines) for a portion of the rail network. The primary study area, from Nashville, TN, to Chattanooga, TN, is bounded by the red dashed line. There are 14 distinct sections of multiple track in the study area. Four OS-points at North and South Murfreesboro and North and South Cowan are also shown on the map, each of which corresponds to a point at which the ETA to the rail yard in Chattanooga is updated.

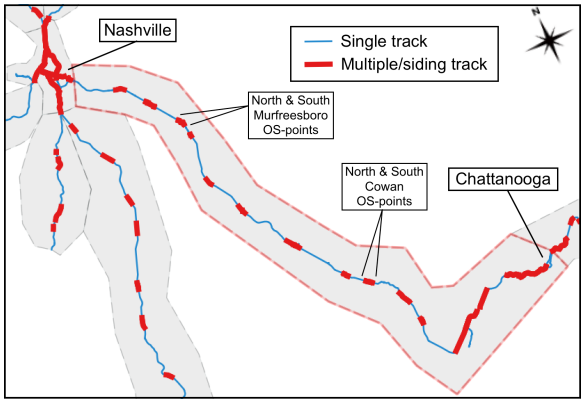


Figure V.3: GIS network view depicting a portion of the Nashville division, with multi-track segments shown in bold red lines and single-track segments in thin blue lines. The primary study route is bounded by the red dashed line. OS-points at Murfreesboro and Cowan are also shown on the map, each of which corresponds to a point at which the ETA to Chattanooga is updated.

V.3.2 Data cleaning and standardization tasks

A variety of data cleaning and data transformation tasks are necessary to organize the input for any prediction algorithm. With over 10,000 trips on the study route in a two year period, we decide to neglect trips with data completeness issues or data errors instead of devising a scheme to impute values; this resulted in the discarding of approximately 10% of trains. Errors consist of fields that contain missing data, or fields that contain illogical values. Examples include non-physical train lengths, or an arrival time prior to the train departure time.

The GIS data is examined to ensure proper connectivity and accuracy before being transformed into the network graph. Common errors encountered include duplicate geometries, disconnected track components, and minor mislabeling of infrastructure components. Many errors are automatically identified and resolved, while some errors require manual correction.

The detection and resolution methods for each of these data fields are summarized in Table V.1. Each is implemented at the time of data mining and feature construction, so that an origin-destination dataset is clean at the time of model training.

V.3.3 Handling of recrewed trains

In the process of early prediction efforts and data exploration efforts, a dominant source of runtime variability has been discovered. Specifically, we find that *recrewed* trains (i.e., a train that did not reach its destination before the crew reached its maximum on-duty time and needed a relief crew) define the dominant source of runtime variability on the study route.

To further investigate the impact of recrews on train variability, all trains are ex post facto labeled as

Table V.1: Data cleaning and standardization steps

Data field	**Data error**	**Data correction**
Train arrival time	arrival time $\leq$ departure time	discard train
Train length	length ≤ 0	discard train
Train tonnage	tonnage ≤ 0	discard train
Train horsepower	horsepower ≤ 0	discard train
Crew assignment	no crew assigned to train	discard train
Crew on duty time	on duty time $\geq$ train departure time	discard train
Track segments	polylines connect spatially, but not by endpoint ID	detect spatial connection, resolve endpoint ID mismatch
Duplicate GIS elements	identical geometry encoded strings	check connectivity of each, keep most connected element

either recrewed or non-recrewed. Less than 10% of the trains on the route were recrewed. The two classes (recrewed and non-recrewed trains) are separated and descriptive statistics are calculated for each class at each of the 35 OS-points, which are spread at irregular intervals across the route depicted in Figure V.3. The standard deviation of runtimes is used to quantify the runtime variability of trains in each class as well as the variability of all trains in the dataset (not separated on recrew). As shown in Figure V.4, the runtime variability of the recrewed trains is several times larger than that of the non-recrewed trains across all OS-points, which are ordered from Nashville to Chattanooga; runtime variability is expressed as a relative value to protect proprietary operational properties in the data. Despite recrewed trains representing less than 10% of the trips, they represent 53% of the variability within the dataset of all trains, when averaged across the full route.

Recrewed trains introduce high variability in runtime and their runtimes are not predictable by features inside the scope of the train and network state features (e.g., status of crew pools from which to reassign and the locations and availability of taxis to transport the crew to the train are significant factors and are not in our dataset). It is not a good idea to include the recrewed trains in the dataset because they have extreme delays, and predicting the duration of the delays is not possible without additional features (e.g., location of the replacement crew). If the recrewed trains are included in the training data, the model will be harder to train because the large error caused by the few but extreme outliers cannot be reduced under any parameter set of the model. Of course, if an outlier robust machine learning algorithm were used in place of the SVR approach, it would be possible to leave the recrewed trains in the training dataset, where they would be effectively ignored. It is likely, however, that the circumstances leading to a recrew will enable its preemptive

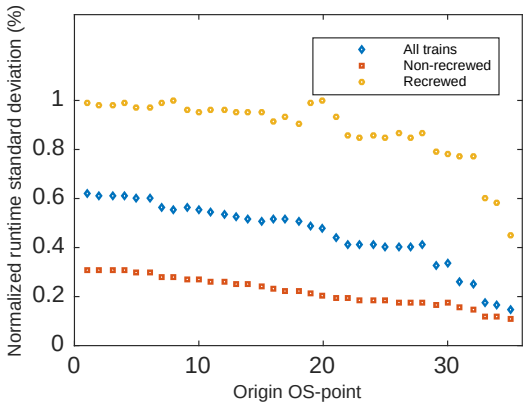


Figure V.4: Comparison of variability in runtime, between recrewed trains, non-recrewed trains, and all trains, for each origin OS point, ordered from Nashville to Chattanooga.
Standard deviation is normalized by the largest variance OS-point, OS-point #21. OS-point IDs increase from Nashville (1) to Chattanooga (42).

classification and could be captured with the available data. It should be noted that features calculated from crew information contain the implicit knowledge that the train was not recrewed, given that the training data was cleaned of recrewed trains; this fact further motivates the need for preemptive classification of recrew events.

V.3.4 Calculated features

This section lists and discusses the features that are generated from the raw data in Section V.3.1 and used to train machine learning algorithms in ETA prediction. A summary of the implemented scalar features appears in Table V.2. These features include six train characteristics, two features that capture the state of the crew on each train, and multiple scalar features quantifying the network characteristics and traffic. We also propose features to describe the traffic state of the network in the geographic vicinity of prediction as vector quantities. Each element of the traffic feature corresponds to the traffic at a track segment between adjacent OS-points. The network traffic state is quantified in terms of occupancy, direction, and priority; these are summarized in Table V.3. All of the features chosen for exploration are based on extensive discussions with operations research personnel from CSX Transportation.

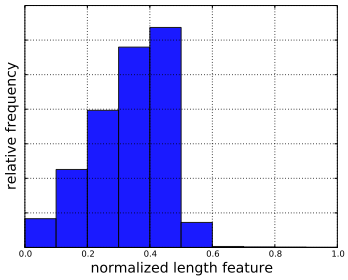
The priority of a train is determined by its cargo (e.g., bulk, merchandise, automotive, intermodal), its type of service (e.g., local, yard, road), and its load status (e.g., loaded, empty). The priority ranking is fully defined for all trains that run on the network and includes exceptions and specialty trains that run infrequently. The priority ranks are aggregated to different levels of granularity. At the highest resolution, all trains are placed into one of twenty priority classes. The relative priority ranking is understood to be non-linear based on its construction. The high resolution ranking is aggregated to a medium-resolution ranking using five

Table V.2: Summary of implemented scalar features.

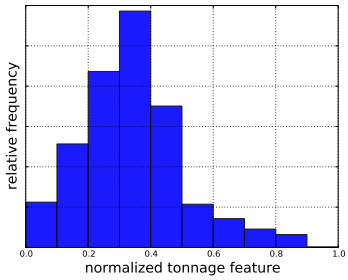
Feature	**Notation**	**Description**
Train length	$\lambda(i)$	The total length of locomotives and cars of train i.
Train tonnage	$\mu(i)$	The total mass of locomotives and cars.
Train horsepower per ton	$\eta(i)$	Total horsepower of locomotives divided by train tonnage, $\mu(i)$.
Train priority (high-resolution)	$\rho_{20}(i)$	Priority ranking on a 1-20 scale.
Train priority (medium-resolution)	$\rho_5(i)$	Five priority classes are constructed by aggregating the high-resolution priority ranking.
Train priority (low-resolution)	$\rho_3(i)$	Three priority classes are constructed by aggregating the high-resolution priority ranking.
Crew time remaining	$\gamma(i)$	Amount of time remaining that the current train crew can legally work.
On duty time to departure	$\theta(i)$	Time between crew on duty time and train departure.
Full traffic count	$\tau(i)$	Count of trains on the remaining line of road.
Directional traffic count	$\tau_\omega(i), \tau_\psi(i)$	Count of trains on the remaining line of road, categorized by direction of travel (i.e., in the same direction, ω, or in the opposite direction of travel, ψ).
Prioritized directional traffic count	$\tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i)$, $\tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i)$	Count of train on the remaining line of road, categorized by both direction of travel and priority relative to that of the train being predicted (i.e., lower or equal priority, β, or higher priority, α).
Available sidings	$\pi(i)$	Count of sidings on route with length greater than that of train i.

Table V.3: List of calculated and implemented track segment feature series, which correspond in dimension to the segmentation of the remaining route, and details for each.

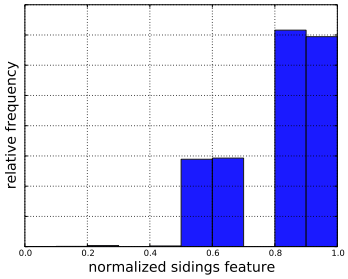
Feature	**Notation**	**Description**
Track segment occupancy	$O(i) = [O_1(i), \cdots, O_l(i)]$	Vector of elements denoting whether a segment on the origin-destination route, indexed 1 to l, is occupied by another train; non-zero when occupied.
Occupying train direction	$D(i) = [D_1(i), \cdots, D_l(i)]$	Denotes the direction (same or opposite) of a train occupying a track segment on the origin-destination route, indexed 1 to l; zero if segment is unoccupied.
Occupying train priority	$P(i) = [P_1(i), \cdots, P_l(i)]$	Assigns high-resolution train priority values to a train occupying a track segment on the origin-destination route, indexed 1 to l. Higher value for high priority train; zero if no train.
Relative priority of occupying train	$R(i) = [R_1(i), \cdots, R_l(i)]$	Non-zero when a train occupying a track segment on the origin-destination route, indexed 1 to l, has higher priority than the train for which the ETA is being predicted. Reflects likelihood of meet/pass delay.
Track segment occupancy around origin point	$G(i) = [G_{-1}(i), \cdots, G_{-h}(i)]$	Indicates track segment occupancy for segments -1 to $-h$ around the origin point, but not included in the primary route (segments 0 to l); captures trains that may enter the primary route and pass/overtake.
Track segment occupancy around destination point	$E(i) = [E_{l+1}(i), \cdots, E_{l+p}(i)]$	Indicates track segment occupancy for segments $l+1$ to $l+p$ around the destination point, but not included in the primary route (segments 0 to l); captures trains that may enter the primary route and conflict during the trip.



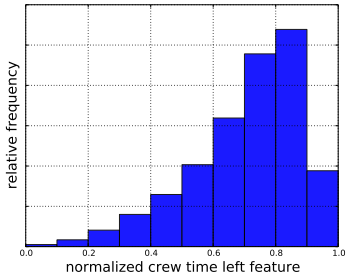
(a) Distribution of train lengths.



(b) Distribution of train tonnage values.



(c) Distribution of number of remaining sidings in which each train will fit.



(d) Distribution of crew time remaining for active train crews.

Figure V.5: Frequency distributions of various scalar features in the training dataset taken at OS-point #1, outside of Nashville.

priority classes and a low-resolution ranking of three priority classes. For example, scheduled merchandise trains have significantly higher priority than bulk/unit trains (e.g., loaded coal train), but in medium- and low-resolution classifications, the two types will get the same priority designation. The priority ranking and aggregations were provided by CSX Transportation.

The physical train characteristics such as train length and train tonnage are calculated by examining the work data, which contains the train dimensions after the most recent work was completed. The distributions of these parameters (shown in Figures V.5a and V.5b) demonstrate that there is a preferred maximum threshold for train length, but that train tonnage is subject to a significant tail of very heavy trains. Train length (together with the track geometry) factors into train runtime, in part because it determines the number of sidings in which a given train may fit. While the majority of sidings are sufficiently long for any train, some sidings are too short to accommodate the longer trains. This disparity is reflected in Figure V.5c, where it is shown that the majority of trains are able to use 80-100% of sidings, while some trains fit into only 50-70% of the sidings.

The most recent crew change can be identified in the crew data, which is then used to calculate the *crew time remaining* (i.e., the maximum time the current crew may continue to operate the train). The distribution of the crew time remaining is shown in Figure V.5d. Ideally, this parameter should be maximal when starting

a route to give the best chance of completing the trip without needing a replacement crew. Note also in Figure V.5d that all trains used to construct this distribution are not recrewed. When combined with the expected train runtime, it is possible to compute a crew *slack time* (i.e., the difference between the crew time remaining and the expected runtime). If the slack time is negative, but is not recrewed, it means the train ran the route faster than the average train. As the slack time becomes more negative, it is increasingly likely that the train will need to be recrewed. In terms of feature construction, the crew time remaining feature and the slack time feature are equivalent under min-max normalization (which is applied to all features in the dataset). Consequently, only the crew time remaining feature is used in the models presented in this work.

It is expected that the traffic along the route influences the runtime, and consequently we propose several methods to quantify the traffic. First, we construct six scalar measures of traffic, each of which consider only the track along the route between the current location of the train and the destination. The six measures differ in the degree of granularity. For example, in the first two measures, we count all other trains (including local trains), which are categorized based on their direction of travel (e.g., same direction, subscript ω, or opposite direction, subscript ψ, relative to the train being predicted). We also consider the fact that the priority of the traffic may also influence the prediction. Consequently we propose four traffic features that enumerate the directional and prioritized traffic counts ($\{$same, opposite$\}$ and higher priority, subscript α, or $\{$same, opposite$\}$ and lower/equal priority, subscript β, where the priority is relative to the train being predicted). It should be noted that the traffic considered in these counts, and in the more granular segmented traffic features, is based on all trains on the network (e.g., including local trains and recrewed trains).

We further examine the potential that the precise location of the traffic may improve the prediction accuracy of the models by considering a higher dimensional representation of the traffic. In the most basic treatment, we can treat each of l track segments between the train and the destination as an element in the feature vector, which is zero if no train is present on the segment or 1 if a train is present. Consequently the dimension of the track occupancy feature is equal to the number of track segments that are considered. For example if no trains are present between the present train and the destination, a vector of zeros of length l (one dimension per segment) would capture the traffic state.

In addition to the track segments between a train origin point and the destination, we consider the track segments in the area around the origin (h track segments) and around the destination (p track segments) that are not part of the origin-destination route segments (l track segments, as stated earlier). This results in the segments from origin to destination being indexed 1 through l, segments around the origin indexed -1 to $-h$, and segments around the destination indexed $l+1$ to $l+p$. The area considered around the origin and around the destination is limited to a distance of 50% of the origin-destination route length, which determines the quantities h and p. On our study route of approximately 140 miles, we consider track segments beyond the

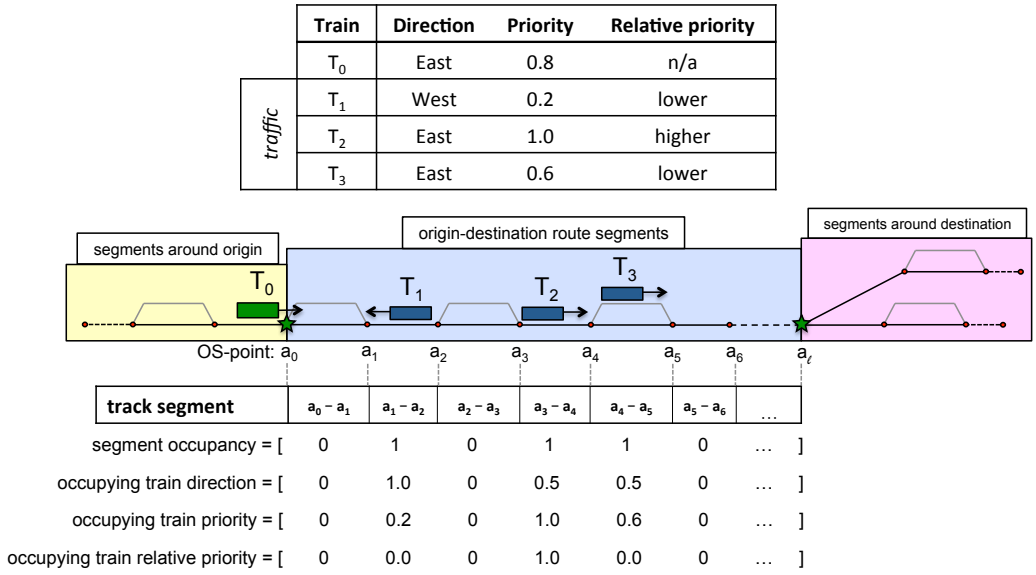


Figure V.6: Segment-wise features are calculated for the area around an origin point, on the origin-destination route, and around the destination. Each is segmented by OS-points, a_0 through a_l on the origin-destination route in this case. The occupancy feature is first constructed, followed by a vector corresponding to the properties of the occupying train when it is present.

destination within 70 miles and track segments around the origin within 70 miles, each excluding the track segments on the 140-mile origin-destination route.

Segment occupancy can be encoded as a vector, each element of which is non-zero when another train is present in a track segment at the time that a prediction is made for a train at the origin point. Likewise, trains that are present on these segments can be described with respect to their relevant properties, namely direction of travel, priority, and relative priority. Each track segment vectors composed of elements that are strictly zero for unoccupied segments and positive for segments occupied by other trains. This process of describing the traffic state via network segments and traffic properties is illustrated in Figure V.6, for origin-destination route, only. Predictions are made at the origin OS-point a_0 and OS-points delineating segments are labeled a_0 through a_l. An example traffic scenario for the moment at which a prediction is made for train T_0 at the origin is shown with trains T_1, T_2, T_3. The relevant features (i.e., direction and priority) of each train are listed in the figure table. The features for trains T_1, T_2, T_3 are mapped to the track segments corresponding to the location of each train resulting in four feature vectors for segment occupancy, occupying train direction, occupying train priority, and occupying train relative priority.

V.4 Model implementation and evaluation

This chapter describes a set of experiments and a metric to assess the machine learning framework described above. The feature sets used in the models are composed of the features described previously in Section V.3.4.

V.4.1 Description of single origin-destination models

Numerical experiments are performed with concentration on a single route, shown by the dashed area in Figure V.3, in the Nashville division of the CSX Transportation network. The network contains a mix of single and double track segments, highly heterogeneous traffic, and high volume relative to capacity. Over 50 distinct trains can be seen on the route in a day and more than 20 of these will typically traverse the full route. The route represents one of the most challenging segments on which to estimate ETAs within the CSX Transportation network. Without loss of generality of the methods, the present analysis is restricted to common train types with sufficient trips in the two year dataset and includes the automotive, merchandise, and intermodal trains. These train types have differing priorities, and consequently have distinct behaviors in meet/pass movements and when delays occur. The dataset for trains running the full study route in the correct direction of travel initially contains over 10,000 trips. When the dataset is filtered by train type, recrewed trains are removed, local trains and trains with intermediate work are eliminated, and data errors and incomplete records are removed, there are still approximately 4,200 trips.

The selected route is composed of 35 points along the 140 mile route for which an ETA to the destination must be produced. For each of the 35 ETA problems, a total of five models are implemented and compared. The models include the baseline median predictor algorithm as well as four SVR-based algorithms. Many combinations of algorithm type and feature set have been explored, and the presented models are representative of the model type and performance. For example, the various priority features have each been evaluated for predictive performance by performing single-feature model experiments and $\rho_5(i)$ is found to be the most informative.

The exact model configurations are as follows:

- Model 0: baseline median predictor where $f(x(i)) = \text{median}\{y(i) \mid y(i) \in Y_{tr}\}$
- Model 1: linear SVR with all scalar features (length, tonnage, hp/ton, priority, crew time, ODTOD, traffic counts, and sidings fit); the feature vector is constructed as: $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i), \tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i)]$, and where $x(i) \in \mathbb{R}^{14}$.
- Model 2: linear SVR with all scalar features plus track segment occupancy vector $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i), \tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i), O_1(i), \cdots, O_l(i)]$, and where $x(i) \in \mathbb{R}^{14+l}$.
- Model 3: linear SVR with all scalar features and all track segment traffic vector quantities (occupancy, direction, priority, and relative priority on origin-destination route; occupancy around origin point; occupancy around destination point) $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i),$

$\tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i), O_1(i), \cdots, O_l(i), D_1(i), \cdots, D_l(i), Q_1(i), \cdots, Q_l(i)$, $R_1(i), \cdots, R_l(i), G_{-1}(i), \cdots, G_{-h}(i), E_{l+1}(i), \cdots, E_{l+p}(i)]$, and where $x(i) \in \mathbb{R}^{14+4l+h+p}$.

- Model 4: RBF kernel SVR with all scalar features and all track segment traffic vector quantities $x(i) = [\lambda(i), \mu(i), \eta(i), \rho_5(i), \gamma(i), \theta(i), \tau(i), \tau_\omega(i), \tau_\psi(i), \tau_{\omega,\alpha}(i), \tau_{\omega,\beta}(i), \tau_{\psi,\alpha}(i), \tau_{\psi,\beta}(i), \pi(i)$, $O_1(i), \cdots, O_l(i), D_1(i), \cdots, D_l(i), Q_1(i), \cdots, Q_l(i), R_1(i), \cdots, R_l(i), G_{-1}(i), \cdots, G_{-h}(i)$, $E_{l+1}(i), \cdots, E_{l+p}(i)]$, and where $x(i) \in \mathbb{R}^{14+4l+h+p}$.

V.4.2 Description of unified all-origin model

The unified all-origin model uses the same feature set as that used by the individual origin-destination models, described in Tables V.2 and V.3. The feature set is therefore a mix of categorical, binary, and continuous quantities. Data records for trains at all locations on the network segment with respect to a single destination are used in the unified model dataset. The selected prediction origin location is included as a one-hot encoding of the route locations.

The resulting feature space has 184 dimensions and the number of labeled data records is over 170,000, each of which represents a feature vector captured at a timing point for the train and the corresponding runtime label. The training and testing data is min-max normalized before being used in the models.

V.4.3 Model evaluation

The error metric used to evaluate a given model and to select model hyper-parameters is *mean absolute error* (MAE), defined as:

$$MAE = \frac{1}{m_{te}} \sum_{i=1}^{m_{te}} |f(x(i)) - y(i)|, \tag{V.7}$$

where $f(x(i))$ and $y(i)$ correspond to the predicted runtime and true runtime of train i, respectively, and m_{te} denotes the number of records in the testing dataset. It follows that numerically low MAE *scores* are better than high scores. Under MAE, all prediction errors are treated equally, regardless of the corresponding true runtime. Performance of each model is compared to that of the historical median predictor, Model 0. The improvement for each model is given as the reduction in the MAE relative to the historical median predictor.

The neural network models were implemented using Keras (Chollet et al., 2015) with TensorFlow (Abadi et al., 2015) backend. Support vector regression, random forest, and statistical models were built using Scikit-Learn (Pedregosa et al., 2011). All models were tested on a computer with 16-core 3.4 GHz processor, 64 GB of RAM, and Nvidia GTX 1080 GPU. Note that neural network models were run on the GPU, while other models were run on the CPU. Neither the preprocessing nor the model testing are parallelized, but could easily be implemented as such on a larger scale due to dataset and model independence.

The data processing steps are completed once for each origin-destination pair and the feature set is stored in a database, which takes approximately 5 minutes per dataset. Building the feature set is the most time consuming step in the process, due primarily to the size of the database of raw data. Adding more features or new data does not require reprocessing of previous data. Model experiments are performed by loading the feature set, selecting the desired data, normalizing features, and training the model and testing the performance via cross-validation. Model training is accomplished in approximately 1 minute for the 4,200 trips, with prediction on test data taking 0.005 seconds per prediction. If implemented network-wide, the computation requirements scale linearly with the number of origin-destination ETA predictions due to trivial parallelization, and could be handled on any modern distributed computing platform.

V.5 Results for individual origin-destination models

This chapter first presents the process for choosing hyper-parameters C and ε in (V.3) for a single origin-destination model with scalar features. We then analyze the series of linear models trained with scalar features on the full 35-OS-point route and the differences between them, specifically with respect to feature weights. Performance results are then shown for each model in Section V.4.1 across the route, which demonstrate the impact of feature richness and the nonlinear kernel.

V.5.1 Choosing hyper parameters for a single model

For each origin-destination model, the SVR parameters, C and ε are chosen as a result of the training and testing process. This is performed using a dataset containing approximately 4200 trips using a five-fold cross validation with an 80/20 training/testing data split. On each fold of the cross validation, the parameter space is explored using a grid search that explores all combinations of parameters within the bounds of each. The results are aggregated across the five folds and the optimal parameter combination is chosen based on the mean minimum testing error. The trained model must be checked for suitability such that it generalizes well to testing data. This check is done using validation curves for C and ε and a learning curve for the amount of data used to train the model.

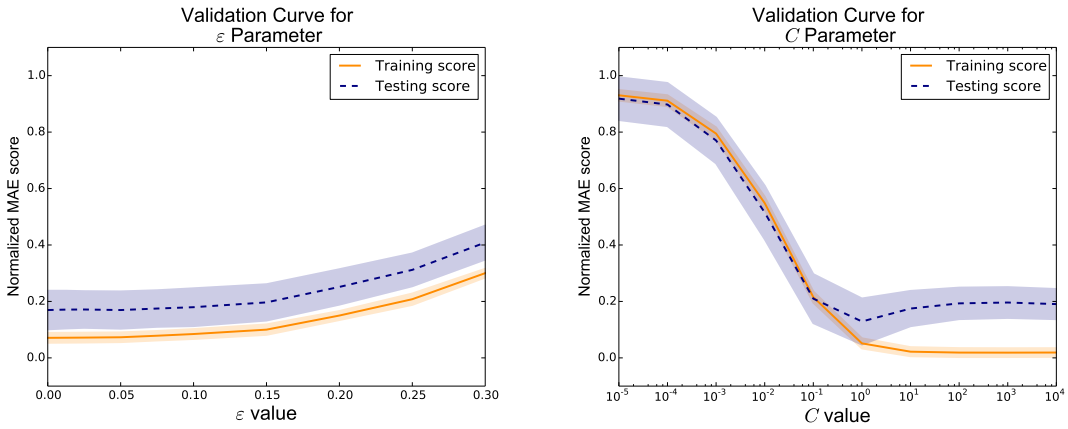
The interpretation of these parameters as well as analysis of training and testing behavior kept the search space limited. The ε parameter is directly related to residual values between $f(x(i))$ and $y(i)$ and, therefore, can be limited to a search space proportional to the normalized spread of the true outputs $y(i)$. The C parameter penalizes the model training error, summed across all observations $x(i)$, relative to the model flatness, given by the two norm of feature weights. We normalize C such that it is scaled by the number of features and inversely scaled by the number of observations in the training dataset. This maintains the impact of the parameter across models with different feature dimensions and data dimensions.

A validation curve explores training and testing scores across a range of a model parameter, with other parameters fixed. Using a fixed C value of 1, the validation curve for the ε parameter in Figure V.7a shows relatively little effect of ε value on training and testing scores. Mean training scores are plotted in the solid orange line, with the narrow shading showing a single standard deviation above and below the mean of training scores in cross-validation. The mean testing scores and distribution of scores are plotted with the dashed purple line and shading. Within the acceptable parameter range, high values nor low values make an appreciable effect on testing MAE. Approaching $\varepsilon = 0$, the ε-insensitive loss disappears and the algorithm converges to linear regression. When ε is set too high, the minimization of prediction errors focuses only on observations with exceptionally high residual prediction values. In comparison, the value of C has significantly higher impact on model score. The validation curve is also shown in Figure V.7b with ε fixed at 0.1 and plotted with common normalized MAE score for comparison. Small values of C emphasize model flatness, but result in poor training and testing scores because model complexity is low. Large values of C achieve low training MAE but generalize poorly to the testing data because of overfitting.

Validation curves show sensitivity for individual parameters. The optimal parameters are chosen simultaneously by evaluating the model on the grid space of all parameter combinations ($C \in [10^{-5}, 10^{4}], \varepsilon \in [0, 0.3]$). For the origin-destination model at grade crossings closest to OS-point #1, the optimal values that minimize MAE are found to be $C = 0.75$ and $\varepsilon = 0.05$. Under these parameters, the difference between the training and testing scores is less than 3% of the training error, which indicates that the model is not overfitting the training data.

The learning curve for a model shows the convergence of training and testing performance by increasing the amount of data available to build the model. The curve is analyzed after hyper-parameters have been chosen. The learning curve shows the MAE score against the number of training examples available to the model. With smaller amounts of data available, training scores will be improved, but at the expense of the model generalizing poorly to testing data. The learning curve in Figure V.8 indicates that the model is trained on a sufficient amount of data because the curves converge before the full training dataset is used. The mean training and testing scores are denoted by the lines with the shaded areas showing one standard deviation in each direction between cross-validation folds. The model training and testing occur at five equally spaced divisions of available data between 10% and 100%, inclusive (i.e., 10%, 32.5%, 55%, 77.5% and 100%). The disparity in training and testing scores begins at over 40% of the training score when using only 10% of the available data, and decreases to less than 3% when 100% of the available data is used.

For any SVR model with a linear kernel, the feature weights can be interpreted meaningfully in both magnitude and sign. The feature weights are recorded at all cross validation folds and for each origin-destination model to assess the relative impact of each feature. The feature weights are normalized by absolute



(a) (b)

Figure V.7: The validation curves for the ε and C parameters on a single origin-destination model are plotted with a common MAE score axis, which is min-max normalized across both parameters. The sensitivity of the model to ε is relatively low compared to that of C.

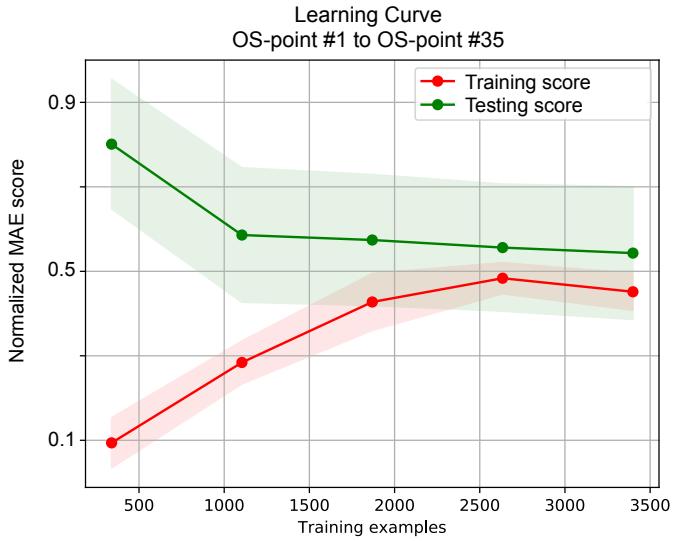


Figure V.8: The learning curve for a single origin-destination model shows convergence of the training and testing error scores given increasing availability of observations in the full dataset. The MAE score values are min-max normalized.

Table V.4: Average feature weights for 5-fold cross validation on origin-destination prediction and Pearson correlation coefficient between feature and runtime, calculated at OS-point #1 (Nashville). Exact model output weights are normalized by absolute sum. Features are ordered by highest mean absolute feature weight.

Feature	**Mean absolute feature weight**	**Pearson correlation coefficient**
Priority, ρ_5	0.346	0.294
Crew time remaining, γ	0.137	-0.124
Tonnage, μ	0.110	0.290
Traffic opposite direction lower/equal priority, $\tau_{\psi,\beta}$	0.089	0.226
Available sidings, π	0.067	0.007
Total traffic, τ	0.055	0.135
Traffic opposite direction, τ_{ψ}	0.050	0.085
Length, λ	0.047	0.014
Traffic same direction, τ_{ω}	0.040	0.121
Horsepower per ton, η	0.019	-0.148
Traffic opposite direction higher priority, $\tau_{\psi,\alpha}$	0.011	-0.114
Traffic same direction higher priority, $\tau_{\omega,\alpha}$	0.010	0.140
Traffic same direction lower/equal priority, $\tau_{\omega,\beta}$	0.009	0.275
On duty time to departure, θ	0.008	0.066

sum within each model, such that $\sum_{j=1}^{n} |w_j| = 1$, to allow comparison between models (recall that the dimension of some traffic features depends on the number of segments between the origin and destination) and are reported in absolute terms for ranking in terms of absolute impact. The feature weight rankings for the origin-destination model at grade crossings closest to OS-point #1 (the beginning of the route in Nashville) are shown in Table V.4. For this model, the effect of train priority is the dominant feature; this is supported by the distinct runtimes between priority classes at this distance from the destination. Crew time remaining has a large impact because it can affect the runtime of a train at this distance if the train experienced significant delay leaving its last terminal. Tonnage also play a large role due to the lower overall performance of the train during acceleration and deceleration. Features with particularly low impact include traffic counts separated by direction and priority, which is an overly simplistic view of the traffic state on long routes. Horsepower per ton also has less of an impact because the train power is typically sized in this region (which contains significant grades) to avoid delays due to under powered trains.

V.5.2 Model training across route

The hyper-parameter selection process and model evaluation is replicated for origin-destination predictions made at each of the 35 OS-points on the full route. Because the C parameter is normalized by the training

data size and feature dimension, and the ε parameter demonstrated little sensitivity, the optimal set of hyper-parameters found by the selection process varied little across the route.

The main finding is that feature weights show significant variability across the route. This supports the idea that dispatching techniques have fundamental differences based on relative location of the train to a terminal point and that unique origin-destination models capture some of this nuance. Noting the important result that the feature weights w learned from SVR are globally optimal and unique (Burges and Crisp, 2000), any change in feature weights from the optimal origin-destination specific weights will result in a loss of accuracy of the predictor.

All scalar feature weights are shown at each prediction point in Figure V.9. The mean feature weight resulting from cross-validation at each location is represented by the lines and min-max ranges for each are shown by the shaded area around the lines. In prediction of the full route, at OS-point #1, the factor most heavily impacting train runtime is priority. Other factors certainly play into the dispatching decisions made for the route, but do not appear to have strong relationships far from the destination. Closer to the destination, traffic counts and train tonnage are driving factors due to decisions around the yard and a natural choke point in the network. The changing importance of train characteristics supports the domain expert knowledge that many factors contribute to train ETA and the impact of these factors is not constant. Along the route, some feature weights experience sharp changes which are due to distinct characteristics of the route. For example, one can observe a sharp dip at OS-point #27 in the weights corresponding to priority and to crew time remaining, along with the sharp increase in the weight corresponding to traffic in the same direction. This OS-point is located on the most significant hill on the route. At this location, a separate helper locomotive attaches to and assists some trains in climbing the hill. Availability of this locomotive is a driving factor in runtime from this location and its presence is captured indirectly in the feature set by the number of trains in the same direction without being explicitly defined as its own feature. The separation of the predictors by each origin-destination allows the estimator to implicitly encode unique attributes of the network which would otherwise require extensive feature engineering.

V.5.3 Performance comparison of SVR models

In this section, the performance of each model detailed in Section V.4.1 is evaluated across all OS-points on the route. The four non-baseline models demonstrate increasing levels of model complexity based on the richness of features used in each model.

The model results across the full route are shown in Figure V.10 in terms of relative reduction in MAE over the historical median predictor (Model 0). A features set with only scalar features (Model 1), as explored in the choice of hyper-parameters and examination of feature weights in Sections V.5.1 and V.5.2, represents

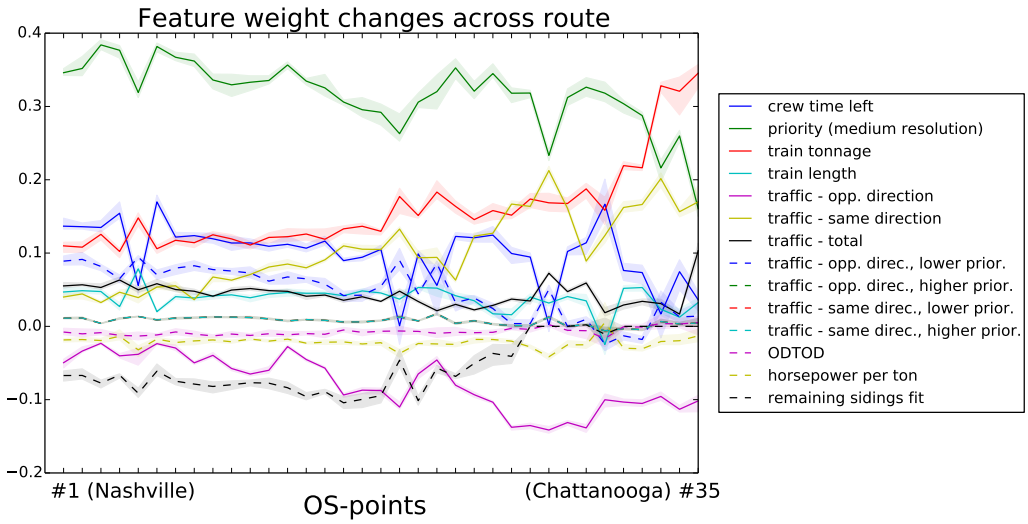


Figure V.9: Feature weight change of scalar features in origin-destination SVR models across the route, Nashville (1) to Chattanooga (35). Feature importance is measured by absolute magnitude.

the largest incremental performance gain for every origin-destination predictor. Inclusion of the track segment feature series (Model 2) and inclusion of all track segment feature series (Model 3) each attain small MAE improvements in addition to improvements gained by the scalar features. The RBF kernel, however, does not provide any substantive performance gain over the fully featured linear model. The γ parameter in the RBF kernel is chosen by exploring the range $\gamma \in [10^{-4}, 10^{3}]$ in a grid search alongside the ε and C parameters.

The scalar feature set (Model 1) contains information representing basic relationships and understandings about how the rail network functions. Unsurprisingly, groups of trains such as those that are heavy and low priority generally run slower than the lighter high priority trains. A few counts indicating the amount of traffic on the route will be roughly indicative of the total amount of delay due to meets and passes. These types of relationships can be determined by a linear model given the information in the scalar feature set. This information would be crucial to any ETA prediction, but it does not provide a complete picture.

Based on the relative performance improvements of the SVR models over the baseline, it is evident that the addition of the network traffic state features (Models 2 and 3) show clear advantages by providing high-resolution information compared to the simple counts of network traffic (Model 1). The larger gains are achieved by the track segment occupancy feature series, but the additional information on the direction and priority of the traffic improve the model performance by better informing on the likely meets and passes that will occur. For instance, the mere presence of another train on the route will be somewhat likely to increase runtime, but if this train is traveling in the direction opposing trains on the origin-destination route and has high priority then it may be highly likely to increase runtime. Moreover, the predictive capability of each

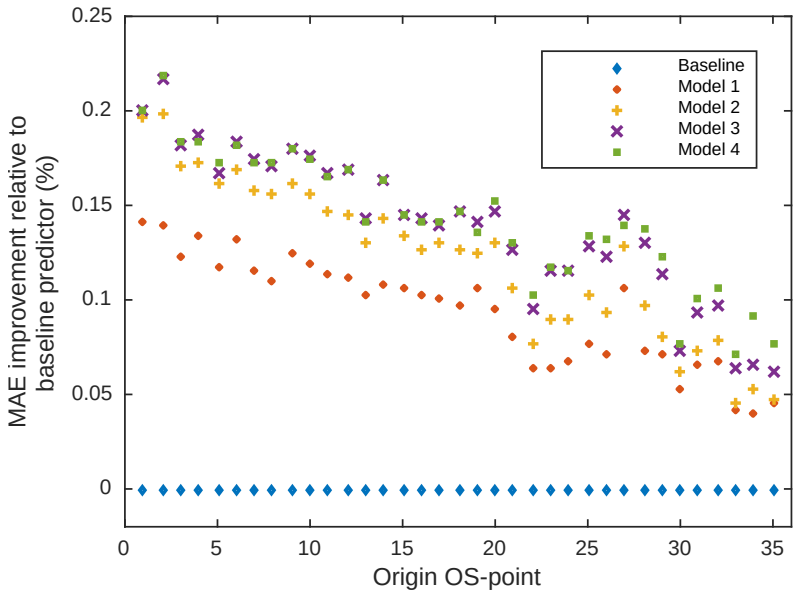


Figure V.10: Improvement in MAE over baseline historical median predictor for each model at all OS-points between Nashville (1) and Chattanooga (35).

type of traffic scenario varies. Generally, ETA error is lower for trains with few potential conflicts than it is for trains with higher numbers of potential conflicts. Additionally, ETA error is lower for trains with large number of conflicting *low* priority trains than it is for trains with large numbers of conflicting *high* priority trains.

Model performance varies somewhat across the origin OS-points due to the distribution of route delay, which is not uniform due to the locations of sidings and the likelihood of each to be used. Overall, the relative performance of the models with respect to each other is consistent across the route. Prediction performance relative to the baseline decreases closer to the destination. We expect this is due to the more unpredictable nature of operations close to rail yards. The factors that affect the exact arrival of a train when it gets close are not necessarily present in the data (e.g., ability of the yard to accept more trains, availability of the next train crew). The magnitude of mean average error for the SVR models follows the same decreasing trend.

These route results are summarized in Table V.5 in terms of mean, maximum, and minimum percent improvement over the baseline. The minimum improvement values are consistently observed for models close to the destination point and the maximum improvement is consistently observed near the beginning of the route.

Table V.5: Comparison of SVR model performance to baseline predictor, summarized for the 35 OS-points on the full route.

Predictor	Mean % improvement	Maximum %	Minimum %
Model 0 (Baseline)	–	–	–
Model 1	9.4%	14.2%	4.0%
Model 2	12.2%	19.9%	4.6%
Model 3	14.0%	21.6%	6.2%
Model 4	14.3%	21.8%	7.0%

V.6 Results for unified all-origin model

In this chapter we explain the tuning process and results for the unified all-origin model and discuss performance.

V.6.1 Choosing hyper parameters for a single model

The SVR model was tuned using an exponential grid space for the hyperparameters C and ε. Kernel hyperparameters (γ for RBF kernel and degree for polynomial kernel) were also tuned in the same grid space. Optimal values were selected from all combinations of hyperparameter values in the grid space and found to be $C = 10.$ and $\varepsilon = 0.075$ for all SVR models. The random forest regression model was tuned by exploring a grid space that included the hyperparameters: number of estimators, maximum features considered in split, and minimum samples required for node split. Values explored in the grid space were chosen based on the dimensionality and characteristics of the data and hyperparameter values were chosen to achieve high predictive performance and minimize overfitting.

The deep neural network model architecture is the results of extensive tuning both in the configuration of hidden layers (from three to ten hidden layers were tested), activation function (ReLU and tanh were tested) and optimization function (Adam and SGD were tested) used in the model. Ultimately, eight hidden layers are used with 200, 200, 150, 100, 70, 40, 20, and 10 nodes in each, respectively. The rectified linear unit (Nair and Hinton, 2010) is chosen for the activation function of neurons; the Adam optimizer (Kingma and Ba, 2014) is found to perform best for training the neural network. Early stopping criteria are employed to avoid overfitting. The learning curve for the resulting Adam-ReLU model is shown in Figure V.11 and compared to the same architecture using stochastic gradient descent optimization. The Adam optimizer not only converges faster (in 32 epochs), but also shows far less variability in validation loss on the way to convergence.

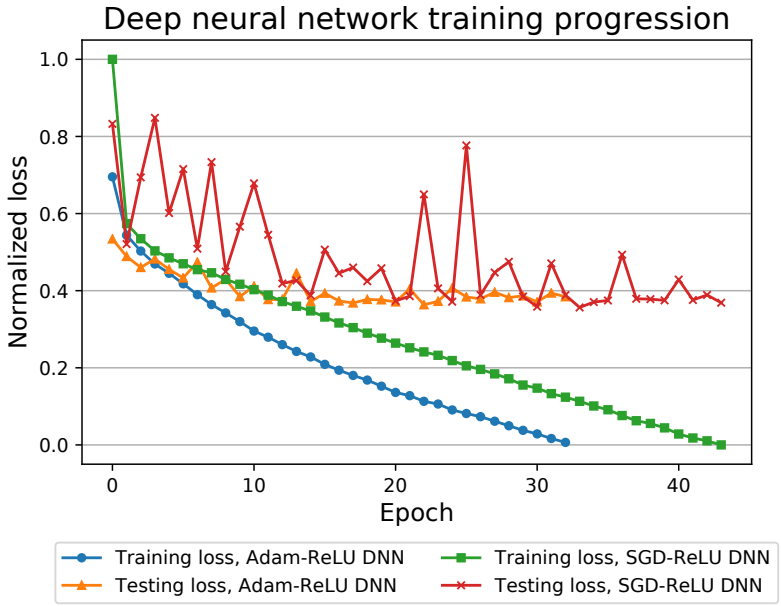


Figure V.11: Learning curve for deep neural network showing normalized training and testing loss values.

V.6.2 Prediction results across route

The predictions resulting from the testing data samples are grouped by the station to which they correspond. Each of these samples serves as an ETA prediction made for a particular train at that station, traveling towards the destination following station 35. The results for all six models are shown in Figure V.12, in terms of percent improvement in MAE compared to the baseline statistical predictor. The results across the route are averaged and shown in Table V.6. Model training times were also monitored and are shown in Table V.7. SVR models are constrained to single-threaded computation in this implementation. Conversely, random forest model can be trained in parallel across CPU cores and the DNN model can use the GPU for computation.

V.6.3 Performance discussion of unified model

There is a noticeable grouping of the SVR models and DNN that maintains over 20% improvement over baseline at stations far from the destination and decreases in relative performance approaching the destination. Linear SVR conspicuously drops below the baseline at the three stations closest to the destination. Meanwhile, the random forest model outperforms all other models at nearly every station. Its performance varies more widely across the route, but achieves improvements exceeding 60% relative to the baseline at stations far from the destination. Predictions made on a longer time horizon are of particular interest for the railroad because of the difficulty of prediction and the increased decision-making potential for hours in the future. The random forest model also see frequent prediction improvements over 50% and average 42%

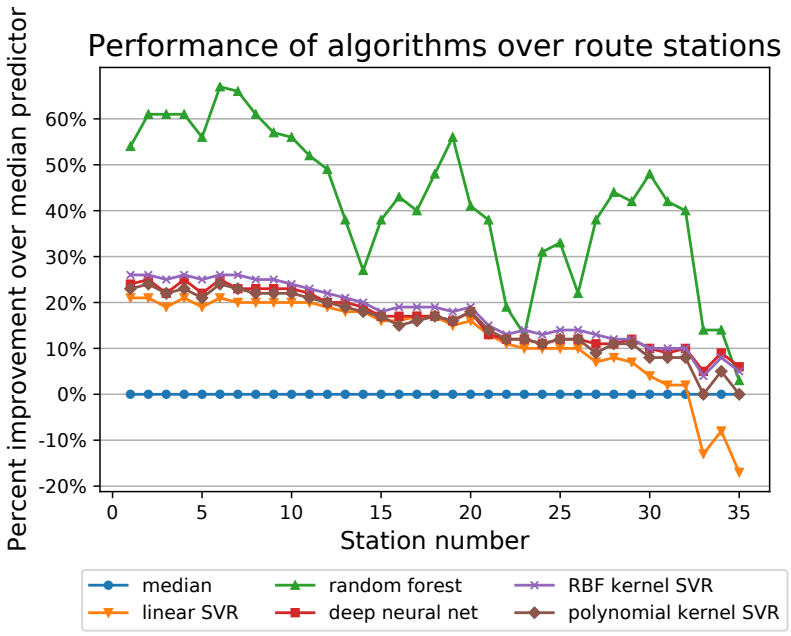


Figure V.12: Relative improvement of arrival time estimates at each station.

Table V.6: Summary of model performance over full route

Model	**Mean % Improvement**	**Maximum % Improvement**
Median	0.0%	0.0%
Linear SVR	12.2%	21.0%
Polynomial kernel SVR	15.3%	23.6%
RBF kernel SVR	17.6%	26.4%
Random Forest	42.1%	67.1%
Deep Neural Net	16.3%	24.9%

Table V.7: Mean model training time.

Model	**Mean training time (seconds)**
Median	0.1
Linear SVR	20
Polynomial kernel SVR	11360
RBF kernel SVR	5560
Random Forest[2]	25
Deep Neural Net[3]	250

improvement over baseline across the route.

As predictions are made closer to the destination, the mean runtime and expected mean average error (in absolute terms) decrease. But mean average error relative to baseline also decreases as predictions were made closer to the destination. This is likely due to the fact that runtimes are also less variable close to a train's destination and the factors that drive the residual variability are difficult to quantify with available data. For example trains can be held outside of the yard due to personnel constraints or space constraints such as lack of availability of a specific track needed (e.g., for refueling or classification). We hope to construct features that quantify this destination yard state in future work.

Fluctuating performance of all models, but particularly the random forest model is notable. This can be explained in part by the nonlinear dynamics of the route. Train and route features differ in their predictive impact by location (Barbour et al., 2018a). For example, route topography plays a role in the predictive impact of train length and tonnage. At locations with a significant hill on the route, long and heavy trains will have a statistically higher runtime than others; but after the hill is traversed, the statistical difference in remaining runtime will diminish. We see a performance variation at approximately the route midpoint that is likely due to a mountain that must be traversed producing an effect of this sort. However, the dramatic performance variability of the random forest model is likely caused by the nuanced relationships that tree-based regressors can extract from categorical and binary data such as the network state used in this work. It is possible that predictions made by the random forest model at some of the low-performing stations depend highly on additional variables not present in the feature space, such as availability of helper locomotives that supplement train power when traversing hills that are present on the route. The training error of the random forest model is up to 10% lower than the testing error (in absolute terms), but the testing results are consistent through cross validation.

V.7 Conclusion

This work presents a data-driven approach to predict ETAs on freight rail networks. The ETA generation problem is posed as 1) a series of independent origin-destination ETA prediction problems to capture location specificity and avoid bias in the training data of a single general model due to time varying features and 2) a unified all-origin prediction model to utilize a greater amount of training data in a single model and leverage more advanced algorithms.

In terms of the origin-destination models, the problem is tractable for sparse rail networks in the United States due to relatively low network complexity that reduces the number of relevant origin-destination pairs. This approach is shown to demonstrate specificity with respect to distinct feature weights (i.e., relative importance) between origin-destination models. Compared to naive prediction based on historical median runtimes,

an average improvement of 14% and maximum improvement of over 21% are achieved by the best performing SVR models.

In terms of the unified all-origin modeling scheme, six models including one statistical model, three SVR models, a deep neural network model, and a random forest regression model are implemented. Performance of the models is analyzed at locations across the study area and found to vary, particularly for the random forest model. The random forest model achieves the best performance yet realized on this dataset, with an average 42% improvement in MAE relative to the baseline statistical predictor. The average improvement of the random forest model and the maximum predictive improvements of over 60% are actionable for freight rail operational decision making.

Based on these findings, our future research steps include the following. Due to the large variance caused by recrews, we are interested in developing a data-driven classifier to preemptively classify trips that are likely to be recrewed. This step is necessary because ETA estimates produced by models trained on data from non-recrewed trains will not generalize to recrewed trains because the ETA of recrewed trains depends on factors outside the scope of this work. For the trains that are not likely to be recrewed, further improvements in the ETA accuracy are possible with the construction of additional targeted traffic features constructed for route topography and meet-pass events. Considering externalities such as weather, as well as ancillary operations such as scheduled track work or slow orders, may also increase prediction accuracy. We are also interested in building models on the state of the origin or departure yard, which may create delays that cascade onto the line of road. Further studying the effects of each input feature to the prediction result may provide practical insight into delay causes and mitigation. The comparison of these results to optimization-based simulation as well as the exploration of more sophisticated deep learning approaches are also areas of future exploration.

This work showed improved results for ETA estimation compared to the state of practice, which can be valuable in rail operations. In implementation, ETA prediction will rely on additional data sources such as real-time GPS data that is collected by the railroads but outside the scope of this work. Further improvements will require model enhancements that incorporate predictive rail traffic evolution and incorporate more of the railroad operational factors that contribute to primary and knock-on delay.

CHAPTER VI

Conclusions and future work

VI.1 Conclusions

This dissertation provides a suite of new data-driven methods that deal with dispatching and prediction on freight railroads in the United States. The main conclusions and findings are as follows:

The *data reconciliation* method for freight rail data automatically cleans dispatching data of infeasibilities and missing values. A formulation is presented for the data reconciliation problem that leverages the same core optimal dispatching constraints to, instead, ingest and clean empirical dispatching data. When applied to a synthetically-decimated dataset for which we have true values of missing points, we find that data reconciliation reduces the timing error of reconstructed data compared to interpolation methods. It also guarantees feasible meet locations for trains and more often correctly identifies meet locations. An objective function utilizing an $\mathcal{L}_2$ norm and average segment speed values is recommended. Historical dispatch data can be populated with small errors for a myriad of reasons, but ultimately, it is critical to ensure the integrity of these datasets when applying them for machine learning or other modeling of freight railroads, which are heavily reliant on data.

The *dispatch analysis problem* and three specific applications of the method answer analysis questions about empirical dispatching decisions. The formulation of the dispatch analysis problem follows, also, from the core of the optimal dispatch problem. By considering empirical network state alongside an optimally-dispatched plan, we pose three questions that can be answered using the method. 1) A lower bound on total train runtime is constructed, when considering a current network state at a point in time, plus optimal replanning into the future. This is shown to isolate specific periods of time in which dispatching added significantly to train runtime, or caused future plans to do so. 2) Alterations to a given network state are found that reduce the lower bound total train runtime in the replanned dispatch. The solution is found to be a sparse set of alterations that highlight particularly impactful alternatives to empirical actions. 3) Dispatch performance of individual trains is quantified, with respect to the train's own effects on the overall runtime of the dispatch plan and its secondary effects on the plan for other trains. Some trains were identified to cause an outsize impact on others, even with a very small difference between their own optimal plan and their actual performance.

Prediction of train estimated times of arrival (ETA) on rail networks can be accomplished within a machine learning formulation. The machine learning problem of generating ETAs for individual freight

trains is posed first a single origin-destination modelling problem and solved with support vector regression (SVR). Clear distinctions, in terms of the driving data features for prediction, are found when models are compared from various origin points to the same destination point on a network segment. A unified origin-destination model is then posed, which considers the dataset of each train's performance across an entire route and makes predictions for new trains from this single model. Random forest regression is shown to out-perform a deep neural network and SVR in this model form and reduce ETA error by an average of over 40% compared to baseline statistical predictors.

VI.2 Strengths and limitations of methods

Data reconciliation

Strengths: The data reconciliation method is extensible, in principle, to any optimization-based rail dispatching model. It is a reformulation of the objective function for the optimal dispatch problem and requires only the inclusion of additional parameters that are the empirical data.

Limitations: The method can correct and impute data only at the fidelity of the underlying model. In this work, the method is discussed only at the track segment level. This is effectively the lowest at which dispatch data could be considered; any less information would not sufficiently encode which track a train used and where meet/overtake events occurred. An example of increased spatiotemporal dispatch data resolution would be GPS data. This data stream could be used to more completely describe train movements with respect to track usage, timing, and speed. This data would allow an optimization-based dispatch model to accurately capture the effects of the end of trains, consider near-instantaneous train speed instead of traversal time of track segments, and more. If a complete reformulation of the model is not desired, given an increased resolution of data, the data may always be aggregated back down to the OS-point level.

Dispatch analysis

Strengths: The dispatch analysis method and process are capable of being reformulated to another optimization-based rail dispatching model because it reformulates the objective functions, adds a few constraints, and adds optimization model parameters to consider empirical data. A straightforward extension would be to a double-track dispatching model, which may use different constraint formulations but have very similar structure.

Limitations: Similarly to data reconciliation, dispatch analysis is presented in this work only at the track segment level. Higher fidelity data could be incorporated, provided an alternative model is capable. Additionally, the dispatch analysis model does not account for higher-fidelity elements of rail dispatching such as train performance, which may be necessary to discern whether or not alternative schedules, replanning, or network state alterations were, in fact, possible.

ETA prediction

Strengths: Machine learning ETA prediction is presented in two model forms, each of which have distinct advantages and drawbacks. Both model formulations, single-origin single-destination and multiple-origin single-destination are viable for predicting the ETA of trains on a short network segment connecting major destinations (e.g., Nashville – Chattanooga). A prediction across a longer route is possible by composing ETA predictions for each smaller segment in succession, but it is likely that long-horizon predictions will rely more heavily on factors not explored in this work and will suffer in accuracy since the network state will evolve significantly.

Limitations: Only the single-origin single-destination model can predict between any two arbitrary points on the network; these points can include a major destination as an intermediate point (e.g., Nashville – Chattanooga – Atlanta). The reason for this discrepancy is that the single-origin single-destination model uses data from any trains that made the trip between the origin and destination using the same path. The train, route, and traffic characteristics that make up the feature space apply over this elongated path, just as they do over a shorter one; though, the feature space may be inadequate to capture the nuanced functions of a path through a major yard/terminal (e.g., crew change, car switching, inspection). Additionally, a large amount of data is needed for the ETA prediction problem, given the intricate relationships between train, network, and traffic characteristics in the features space. For less-frequent origin-destination predictions, the amount of data may be insufficient to successfully train a model.

VI.3 Proposed future work

In future work, I intend to continue the examination of detailed rail dispatch data to identify further strategies for examining rail performance and enhancing modeling and prediction of freight rail transportation.

To further examine the impact of dispatching decisions I would like to extend problem 1 of the dispatch analysis applications to isolate primary and secondary effect of individual dispatch actions. Similar to the separation of added runtime effects for individual trains, separating the primary and secondary effects for individual actions can reveal how a disturbance or delay which introduced only small primary effects could cause greater secondary effects on future actions.

I believe that optimization-based dispatching can also be a useful component of prediction. Similar to the development of ETA machine learning predictors, a set of train features and network state could be used in conjunction with optimal dispatching to predict the future evolution of the network and the arrival times of individual trains. I intend to explore this idea in future work, as well.

Appendices

Appendix A

ETA predictions at grade crossings

A.1 Grade crossing arrival times

In 2015, there were nearly 3,000 collisions between vehicles and trains that resulted in approximately 230 fatalities at *grade crossings*, locations where roadways and pathways cross railroad tracks at grade. In the United States today, there are 216,000 grade crossings (Federal Railroad Administration, 2018). Grade crossings are the second highest contributor to rail-related fatalities, after trespassing (Federal Railroad Administration Office of Safety Analysis, 2018); these two causes cover over 90% of rail-related fatalities.

Grade crossings are not just problematic because of safety concerns related to collisions with vehicles. Additionally, occupancy time at grade crossings can be large, which causes congestion on surface streets and, notably, delays and blockages for emergency vehicles. Notably systematic problems have been observed where congested tracks lead to grade crossing blockages of multiple hours (Surface Transportation Board, 2016). Legal disagreements have occurred between state and local entities attempting to assert control over rail grade crossings, but control has rested at the Federal level for train movements (Wronski, 2008). This is true even at grade crossings with roads owned and operated by cities and states.

The Federal Railway Administration (FRA) has undertaken research on the use of intelligent transportation system technology at grade crossings to enhance safety and warnings by providing more complete information from positive train control (United States Department of Transportation and Federal Railroad Administration, 2007). This technology framework, shown in Figure A.1, was formalized in IEEE Standard 1570-2002 (IEEE Vehicular Technology Society, 2002). Emergency vehicle preemption of road signals is also well-studied and has shown to be effective and have minimal negative impacts, but emergency vehicles do not have the ability to preempt or otherwise influence rail operations (Nelson and Bullock, 2000). Any proactive intervention framework requires that *estimated times of arrival* (ETAs) of trains at grade crossings be generated, motivating our present work.

Grade crossing safety is a challenging problem because freight and passenger trains have large stopping and acceleration distances due to their mass and speed. This problem is compounded by shared corridors, high-speed rail, and increasingly long and heavy freight trains (Chadwick et al., 2014). Large freight trains can require up to a mile of emergency stopping distance, well outside the range of what is reasonable to prevent most grade crossing incidents. Additional risks are created by collisions, as they have the potential to cause train derailments (Chadwick et al., 2012).

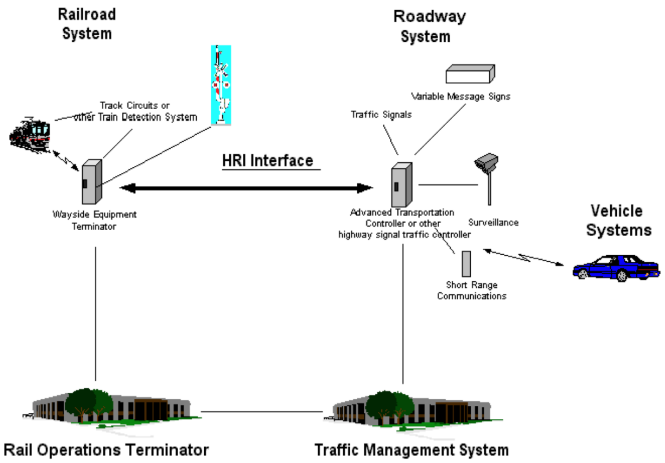


Figure A.1: IEEE Standard 1570-2002 highway-rail intersection interface overview (IEEE Vehicular Technology Society, 2002).

Compliance with grade crossing safety measures is not absolute and enforcement is difficult. Drivers do not have good comprehension of safety marking and devices installed at grade crossings (Richards and Heathington, 1988). Moreover, drivers will fail to notice or obey grade crossing warnings (Meeker et al., 1997). There are numerous means by which to improve driver behavior and, thus, safety at grade crossings. Results from Project Lifesaver have shown a positive effect by reducing collisions at grade crossings through awareness and education (Savage, 2006). Detection and enforcement by video camera has also been an area of study (Kim and Cohn, 2004). Predictive models are able to identify particularly problematic or at-risk crossings from a safety standpoint (Medina and Benekohal, 2015).

Safety at grade crossings has been, and continues to be, a priority of the FRA and the United States Department of Transportation (USDOT). In 2015, the FRA announced a partnership with Google to incorporate grade crossing locations into mapping data that many drivers use to navigate (Tumulty, 2015). Brady (2003) incorporated potentially blocked rail crossings into routing during emergency response and management. Also related is the problem of facility location planning for effective emergency response to incidents that include railroad-related events (Ouyang et al., 2018). These large concerns over grade crossing safety demonstrate the importance of proactively addressing safety at grade crossings with respect to the impending arrival of freight trains. For early warning systems or advanced routing systems to be useful for passenger, commercial, and emergency vehicles, accurate ETAs must be generated for train arrivals at grade crossings. Arrival data at grade crossings is generally not a subject of collection by railroads or public agencies. Additionally, there still exists a technological gap to anticipate train arrivals at grade crossings on a longer time horizon, up to multiple hours. ETA prediction on the necessary longer time horizons requires more than just

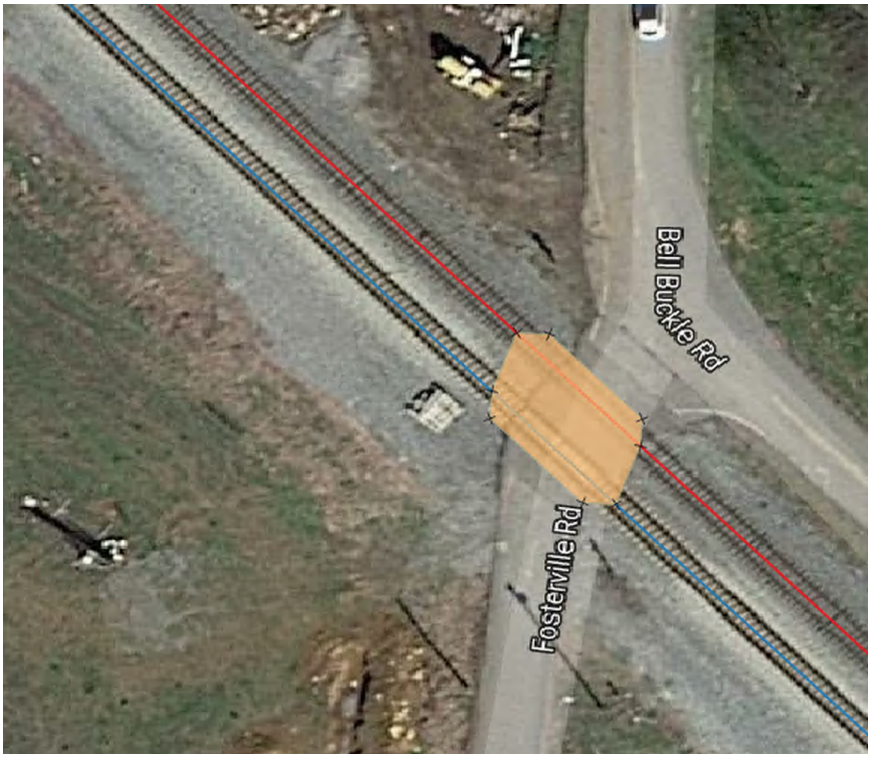


Figure A.2: A single grade crossing boundary delineated by the orange shaded area and overlaid on a satellite imagery basemap. The tracks at this location are shown by the blue line (main line track) and red line (siding track).

real-time positioning of a train. It requires the consideration of each individual train, the larger rail network, and interactions between trains on the network.

We demonstrate and evaluate prediction methods with commonly-used control point timing data from the railroad, because grade crossing arrival data is not available. These control point data represent the best available large-scale data source for U.S. freight rail that has been available for research. Using other data sources, such as GPS or dispatch data, grade crossing prediction models can be constructed equivalently to the models demonstrated for control points. We explain this modeling choice and its implications further in Section A.2 and Section A.3.

Grade crossing raw data is in the form of GIS polygons, which correspond to the track in the coordinate reference system (CRS). See Figure A.2 for an example of one of these grade crossing data points, where the crossing is delineated by the orange boundary.

A.2 Grade crossings data and locations

Grade crossings are frequent elements of the rail network. They are spaced at uneven intervals on virtually every segment that is operated; for example, see a sample of grade crossings outside of Nashville, TN, in

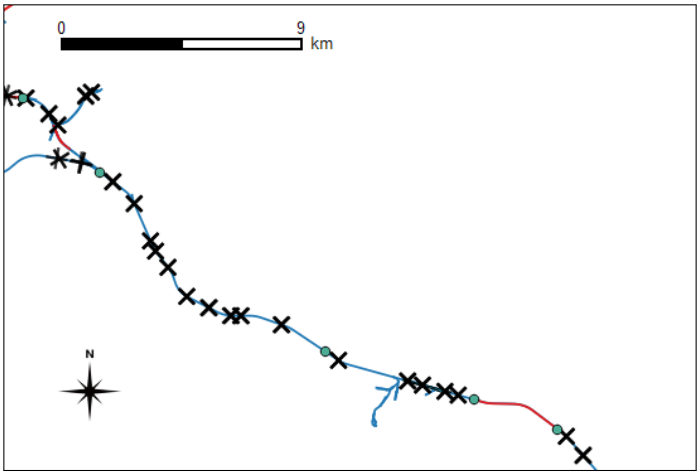


Figure A.3: Sample of grade crossings, denoted by 'x' symbols, outside of Nashville, TN.

Figure A.3. For each grade crossing, data exists on its precise location and extents, as shown in Figure A.2. However, train timing is not reported at precise grade crossing locations, only at nearby OS-points through the dispatching system. Therefore, we associate each grade crossing by minimum distance with an OS-point, at which we have train timing data to train and test models. The number of grade crossings associated with each OS-point between Nashville, TN, and Chattanooga, TN, are shown in Figure A.4, which shows a very uneven distribution.

A.3 Prediction of arrival times at grade crossings

In this section, we explore the problem of ETA prediction to grade crossings. Specifically, we discuss the limitations and difficulties, modeling choice, and results.

A.3.1 Prediction limitations

One of the principal difficulties of making arrival predictions at grade crossings is their frequency relative to the sparsely-distributed timing points on the network (OS-points). This raises the notable limitation that we do not have ground truth arrival times at these locations. Interpolating expected arrival times at grade crossings for use in model evaluation would only increase error and uncertainty. Consequently, we can only judge the validity of ETA's made at the upstream and downstream OS-points. The best arrival time prediction at the nearest OS-point would need to be made and correcting based on the precise location of the grade crossing relative to the OS-point. Additional factors could be relevant in calculating the correction factor, such as the last time the train encountered a meet or pass maneuver, which could alter its speed and acceleration state. In the scope of this work, we assess predictions made only to a grade crossing's nearest OS-point because of the absence of ground truth arrival data at grade crossings. Despite this limitation, model performance is still

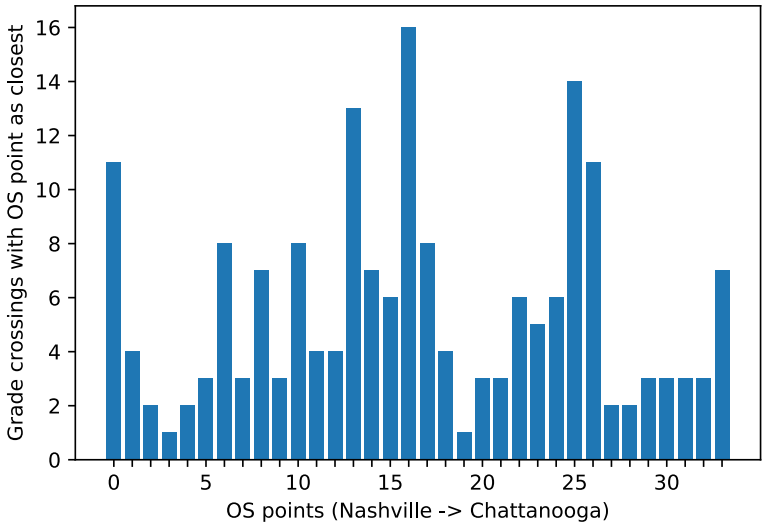


Figure A.4: Number of grade crossings associated by minimum distance with OS-points between Nashville, TN, and Chattanooga, TN.

indicative of that expected precisely at grade crossings, due to the magnitude of distance between OS-point and grade crossing being on the order of a single mile.

This limitation can be addressed using additional data sources that railroads collects that are not available for research. We believe this is the correct scientific approach that provides the community realistic estimates of the achievable accuracy the first study of its kind in the United States.

A.3.2 Model choice and construction

In order to predict arrival times at grade crossings, we use the individual origin-destination modelling scheme. The origin-destination modelling results supported the notion of location-specific models with respect to the origin point. It follows that this location-specificity would be increasingly important when building models with varying origins and varying destinations (grade crossings). We construct models for each two-OS-point pair on the Nashville-Chattanooga subdivision as a predictor of the ETA to grade crossings associated with that OS-point. We consider trains traveling in only the direction from Nashville to Chattanooga and place a lower bound limit on the distance between OS-points due to known variability factors of train runtimes between nearby OS-points for which we can not quantify with data inside the scope of this work.

A.3.3 Performance discussion of grade crossing models

The mean improvement in MAE relative to the baseline predictor is shown in Figure A.5 for predictions made to each OS-point aggregated across other OS-points on the network serving as origin points.

Model performance shows notable variations for predictions made between OS-points as a best proxy for

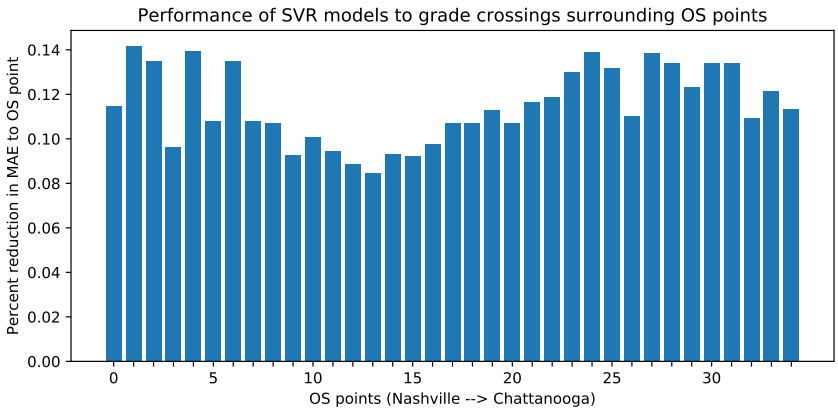


Figure A.5: Mean improvement percentage over baseline statistical model for predictions made to grade crossings associated with each OS-point.

nearby grade crossings. However, the values for improvement in MAE are very comparable to those observed in individual single-destination models across the same track territory. A maximum mean improvement in MAE of 14% was observed at multiple destination OS-points.

A.4 Practical model implementation

As previously discussed, there are limitations in the present work inherent to the accessible data sources. These would need to be addressed as a necessary condition for implementing real-time grade crossing prediction models, but are entirely surmountable. Making simultaneous predictions for thousands of trains at thousands of grade crossings would require a large scale computing environment, but this is a problem well within the realm of technological feasibility for modern distributed computing. As noted in Section V.4.3, model training required a significant amount of time up front and would need to be performed periodically to keep models up to date. But training may be performed offline so as to not interfere with real-time online prediction. Each prediction on test data in this work took on average 0.005 seconds. This is reasonable for a real-time ETA prediction system.

Though the analysis in this work focuses on predictions made on single-track network segments, the same prediction framework is immediately extensible to any network configuration. Models could be constructed of the same form because the nuance introduced by some geographic factors is captured inherently in historical data. We also expect that this approach is applicable for all freight railroads in the United States and, potentially, elsewhere in the world.

Implementation of this work fits into the IEEE Standard 1570-2002 for highway-rail intersection interface (discussed in Section A.1) as a long-horizon train detection system that informs traffic management systems. Ouyang et al. (2018) addressed strategic incidence response planning in the presence of correlated disruptions.

Specifically, where assets and resources should be positioned in anticipation of potential accidents when there is a risk of systematic disruption to the transportation network. They note, in particular, trains blocking railroad crossings as a common example. The joint consideration of our work on real-time prediction with strategic planning could enable robust emergency response operations under the inevitable disruptions.

A.5 Conclusion

This work showed improved results for ETA estimation compared to the state of practice, which can be valuable to emergency vehicle scheduling and management around highway-rail grade crossings. In terms of implementation, the real time up-to-the-minute reporting of arrival times at grade crossings will also require additional data inputs such as real-time GPS data.

BIBLIOGRAPHY

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Retrieved from https://www.tensorflow.org/.

Abril, M., Barber, F., Ingolotti, L., Salido, M. A., Tormos, P., and Lova, A. (2008). An assessment of railway capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):774–806.

Altinkaya, M. and Zontul, M. (2013). Urban bus arrival time prediction: A review of computational models. *International Journal of Recent Technology and Engineering (IJRTE)*, 2(4):164–169.

Amtrak (2016). Route on-time performance.

Assad, A. A. (1980). Models for rail transportation. *Transportation Research Part A: General*, 14(3):205–220.

Association of American Railroads (2013). Class I railroad statistics.

Barbour, W., Mori, J. C. M., Kuppa, S., and Work, D. B. (2018a). Prediction of arrival times of freight traffic on us railroads using support vector regression. *Transportation Research Part C: Emerging Technologies*, 93:211–227.

Barbour, W., Samal, C., Kuppa, S., Dubey, A., and Work, D. B. (2018b). On the data-driven prediction of arrival times for freight trains on us railroads. In *Proceedings of the IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2289–2296.

Basak, D., Pal, S., and Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing-Letters and Reviews*, 11(10):203–224.

Bollapragada, S., Markley, R., Morgan, H., Telatar, E., Wills, S., Samuels, M., Bieringer, J., Garbiras, M., Orrigo, G., Ehlers, F., Turnipseed, C., and Brantley, J. (2018). A novel movement planner system for dispatching trains. *Interfaces*, 48(1):57–69.

Bonsra, K. B. and Harbolovic, J. (2012). Estimation of run times in a freight rail transportation network. Master's thesis, Massachusetts Institute of Technology.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152. ACM.

Brady, T. F. (2003). Public health: emergency management: capability analysis of critical incident response. In *Proceedings of the 35th conference on Winter simulation: driving innovation*, pages 1863–1867, New Orleans, LA, USA.

Breiman, L. (1984). *Classification and regression trees*. Routledge, 1st edition.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Burges, C. J. and Crisp, D. J. (2000). Uniqueness of the SVM solution. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 223–229.

Cambridge Systematics (2007). National rail freight infrastructure capacity and investment study. *Cambridge Systematics, Cambridge*.

Carey, M. and Kwieciński, A. (1994). Stochastic approximation to the effects of headways on knock-on delays of trains. *Transportation Research Part B: Methodological*, 28(4):251–267.

Chadwick, S. G., Saat, M. R., and Barkan, C. P. (2012). Analysis of factors affecting train derailments at highway-rail grade crossings. In *Proceedings of the Transportation Research Board 91st Annual Meeting*, Washington, DC, USA.

Chadwick, S. G., Zhou, N., and Saat, M. R. (2014). Highway-rail grade crossing safety challenges for shared operations of high-speed passenger and heavy freight rail in the us. *Safety Science*, 68:128–137.

Chapuis, X. (2017). Arrival time prediction using neural networks. In *Proceedings of RailLille2017: 7th International Conference on Railway Operations Modelling and Analysis*, pages 1500–1510. International Association of Railway Operations Research (IAROR).

Chen, B. and Harker, P. T. (1990). Two moments estimation of the delay on single-track rail lines with scheduled traffic. *Transportation Science*, 24(4):261–275.

Chien, S. I.-J., Ding, Y., and Wei, C. (2002). Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering*, 128(5):429–438.

Chollet, F. et al. (2015). Keras. Retrieved from https://keras.io.

Claudel, C. G. and Bayen, A. M. (2011). Convex formulations of data assimilation problems for a class of Hamilton–Jacobi equations. *SIAM Journal on Control and Optimization*, 49(2):383–402.

Corman, F. and D'ariano, A. (2012). Assessment of advanced dispatching measures for recovering disrupted railway traffic situations. *Transportation Research Record*, 2289(1):1–9.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Daamen, W., Goverde, R. M., and Hansen, I. A. (2009). Non-discriminatory automatic registration of knock-on train delays. *Networks and Spatial Economics*, 9(1):47–61.

D'Ariano, A. and Pranzo, M. (2009). An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9(1):63–84.

D'Ariano, A., Pranzo, M., and Hansen, I. A. (2007). Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222.

Dick, C. T. and Mussanov, D. (2016). Operational schedule flexibility and infrastructure investment: capacity trade-off on single-track railways. *Transportation Research Record*, 2546(1):1–8.

Dingler, M., Koenig, A., Sogin, S., and Barkan, C. P. (2010). Determining the causes of train delay. In *AREMA Annual Conference Proceedings*.

Dingler, M., Lai, Y.-C., and Barkan, C. P. (2009). Impact of train type heterogeneity on single-track railway capacity. *Transportation Research Record: Journal of the Transportation Research Board*, (2117):41–49.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing Systems*, pages 155–161.

Estes, R. M. and Rilett, L. R. (2000). Advanced prediction of train arrival and crossing times at highway-railroad grade crossings. *Transportation Research Record: Journal of the Transportation Research Board*, (1708):68–76.

Fang, W., Yang, S., and Yao, X. (2015). A survey on problem models and solution approaches to rescheduling in railway networks. *Transactions on Intelligent Transportation Systems*, 16(6):2997–3016.

Federal Railroad Administration (2012). Freight railroad background. Technical report.

Federal Railroad Administration (2018). Highway-rail crossing inventory data. Retrieved from https://safetydata.fra.dot.gov/officeofsafety/publicsite/downloaddbf.aspx.

Federal Railroad Administration Office of Safety Analysis (2018). Safety statistics. Retrieved from https://safetydata.fra.dot.gov/OfficeofSafety/default.aspx.

Furtado, F. M. B. A. (2013). US and European freight railways: The differences that matter. *Journal of the Transportation Research Forum*, 52(2):65–84.

Gestrelius, S., Aronsson, M., Forsgren, M., and Dahlberg, H. (2012). On the delivery robustness of train timetables with respect to production replanning possibilities.

Gestrelius, S., Aronsson, M., and Peterson, A. (2017). A MILP-based heuristic for a commercial train timetabling problem. *Transportation Research Procedia*, 27:569–576.

Ghofrani, F., He, Q., Goverde, R. M., and Liu, X. (2018). Recent applications of big data analytics in railway transportation systems: A survey. *Transportation Research Part C: Emerging Technologies*, 90:226–246.

Ghoseiri, K., Szidarovszky, F., and Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. *Transportation research part B: Methodological*, 38(10):927–952.

Gorman, M. F. (2009). Statistical estimation of railroad congestion delay. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):446–456.

Goverde, R. M. (2005). *Punctuality of railway operations and timetable stability analysis*. Netherlands TRAIL Research School.

Goverde, R. M. and Meng, L. (2011). Advanced monitoring and management information of railway operations. *Journal of Rail Transport Planning & Management*, 1(2):69–79.

Hallowell, S. F. and Harker, P. T. (1998). Predicting on-time performance in scheduled railroad operations: Methodology and application to train scheduling. *Transportation Research Part A: Policy and Practice*, 32(4):279–295.

Hansen, I. A., Goverde, R. M., and van der Meer, D. J. (2010). Online train delay recognition and running time prediction. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1783–1788. IEEE.

Hellström, P., Kauppi, A., Wikström, J., Andersson, A. W., Sandblad, B., Kvist, T., and Gideon, A. (2003). Experimental evaluation of decision support tools for train traffic control. In *Proceedings of The World Congress on Railway Research*, pages 670–677.

Hertenstein, J. H. and Kaplan, R. S. (1991). *Burlington Northern: The ARES decision*. Harvard Business School.

Higgins, A., Ferreira, L., and Kozan, E. (1995). Modelling delay risks associated with train schedules. *Transportation Planning and Technology*, 19(2):89–108.

Higgins, A., Kozan, E., and Ferreira, L. (1996). Optimal scheduling of trains on a single line track. *Transportation Research Part B: Methodological*, 30(2):147–161.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hwang, C.-C. and Liu, J.-R. (2009). A simulation model for estimating knock-on delay of taiwan regional railway. In *Proceedings of the Eastern Asia Society for Transportation Studies Vol. 7 (The 8th International Conference of Eastern Asia Society for Transportation Studies, 2009)*, pages 213–213. Eastern Asia Society for Transportation Studies.

IEEE Vehicular Technology Society (2002). Standard for the interface between the rail subsystem and the highway subsystem at a highway rail intersection. *IEEE Standards*, (1570-2002).

Kecman, P. and Goverde, R. M. (2013). An online railway traffic prediction model. In *RailCopenhagen2013: 5th International Conference on Railway Operations Modelling and Analysis*. International Association of Railway Operations Research (IAROR).

Kecman, P. and Goverde, R. M. (2015). Online data-driven adaptive prediction of train event times. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):465–474.

Khadilkar, H., Salsingikar, S., and Sinha, S. K. (2017). A machine learning approach for scheduling railway networks. In *Proceedings of RailLille2017: 7th International Conference on Railway Operations Modelling and Analysis*. International Association of Railway Operations Research (IAROR).

Khoshniyat, F. and Peterson, A. (2015). Robustness improvements in a train timetable with travel time dependent minimum headways. In *Proceedings of the 6th International Conference on Railway Operations Modelling and Analysis (RailTokyo2015)*.

Khoshniyat, F. and Peterson, A. (2017). Improving train service reliability by applying an effective timetable robustness strategy. *Journal of Intelligent Transportation Systems*, pages 1–19.

Kim, Z. and Cohn, T. E. (2004). Pseudoreal-time activity detection for railroad grade-crossing safety. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):319–324.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kraay, D. R. and Harker, P. T. (1995). Real-time scheduling of freight railroads. *Transportation Research Part B: Methodological*, 29(3):213–229.

Lamorgese, L. and Mannino, C. (2015). An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, 63(1):48–64.

Leibman, M., Edgar, T., and Lasdon, L. (1992). Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers & Chemical Engineering*, 16(10-11):963–986.

Li, H., Parikh, D., He, Q., Qian, B., Li, Z., Fang, D., and Hampapur, A. (2014). Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, 45:17–26.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Lovett, A. H., Dick, C. T., and Barkan, C. P. (2015). Determining freight train delay costs on railroad lines in North America. In *Proceedings of RailTokyo2015: 5th International Conference on Railway Operations Modelling and Analysis*. International Association of Railway Operations Research (IAROR).

Lovett, A. H., Dick, C. T., and Barkan, C. P. (2017). Predicting the cost and operational impacts of slow orders on rail lines in north america. In *Proceeding of the 7th International Conference on Railway Operations Modelling and Analysis (RailLille2017), Lille, France, 4-7 April 2017*.

Lu, Q., Dessouky, M., and Leachman, R. C. (2004). Modeling train movements through complex rail networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 14(1):48–75.

Lusby, R. M., Larsen, J., and Bull, S. (2018). A survey on robustness in railway planning. *European Journal of Operational Research*, 266(1):1–15.

Marinov, M. and Viegas, J. (2011). A mesoscopic simulation modelling methodology for analyzing and evaluating freight train operations in a rail network. *Simulation Modelling Practice and Theory*, 19(1):516–539.

Marković, N., Milinković, S., Tikhonov, K. S., and Schonfeld, P. (2015). Analyzing passenger train arrival delays with support vector regression. *Transportation Research Part C: Emerging Technologies*, 56:251–262.

Medina, J. C. and Benekohal, R. F. (2015). Macroscopic models for accident prediction at railroad grade crossings: comparisons with us department of transportation accident prediction formula. *Transportation Research Record: Journal of the Transportation Research Board*, (2476):85–93.

Meeker, F., Fox, D., and Weber, C. (1997). A comparison of driver behavior at railroad grade crossings with two different protection systems. *Accident Analysis & Prevention*, 29(1):11–16.

Milinković, S., Marković, M., Vesković, S., Ivić, M., and Pavlović, N. (2013). A fuzzy petri net model to estimate train delays. *Simulation Modelling Practice and Theory*, 33:144–157.

Mori, U., Mendiburu, A., Álvarez, M., and Lozano, J. A. (2015). A review of travel time estimation and forecasting for advanced traveller information systems. *Transportmetrica A: Transport Science*, 11(2):119–157.

Mu, S. and Dessouky, M. (2011). Scheduling freight trains traveling on complex networks. *Transportation Research Part B: Methodological*, 45(7):1103–1123.

Murali, P., Dessouky, M., Ordóñez, F., and Palmer, K. (2010). A delay estimation technique for single and double-track railroads. *Transportation Research Part E: Logistics and Transportation Review*, 46(4):483–495.

Murali, P., Ordóñez, F., and Dessouky, M. M. (2016). Modeling strategies for effectively routing freight trains through complex networks. *Transportation Research Part C: Emerging Technologies*, 70:197–213.

Mussanov, D., Nishio, N., and Dick, C. T. (2017). Delay performance of different train types under combinations of structured and flexible operations on single-track railway lines in north america. In *Proceedings of the International Association of Railway Operations Research (IAROR) 7th International Seminar on Railway Operations Modelling and Analysis*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Narayanaswami, S. and Rangaraj, N. (2011). Scheduling and rescheduling of railway operations: A review and expository analysis. *Technology Operation Management*, 2(2):102–122.

Nelson, E. and Bullock, D. (2000). Impact of emergency vehicle preemption on signalized corridor operation: An evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, (1727):1–11.

Oneto, L., Buselli, I., Lulli, A., Canepa, R., Petralli, S., and Anguita, D. (2019). A dynamic, interpretable, and robust hybrid data analytics system for train movements in large-scale railway networks. *International Journal of Data Science and Analytics*, pages 1–17.

Oruganti, A., Sun, F., Baroud, H., and Dubey, A. (2016). Delayradar: A multivariate predictive model for transit systems. In *Proceedings of the IEEE International Conference on Big Data*, pages 1799–1806.

Ouyang, Y., Xie, S., and An, K. (2018). Positioning, planning and operation of emergency response resources and coordination between jurisdictions. Center for Transportation Studies, University of Minnesota.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pellegrini, P., Marlière, G., and Rodriguez, J. (2016). A detailed analysis of the actual impact of real-time railway traffic management optimization. *Journal of Rail Transport Planning & Management*, 6(1):13–31.

Petersen, E. and Taylor, A. (1982). A structured model for rail line simulation and optimization. *Transportation Science*, 16(2):192–206.

Petersen, E., Taylor, A., and Martland, C. (1986). An introduction to computer-assisted train dispatch. *Journal of Advanced Transportation*, 20(1):63–72.

Pouryousef, H., Lautala, P., and White, T. (2015). Railroad capacity tools and methodologies in the US and Europe. *Journal of Modern Transportation*, 23(1):30–42.

Richards, S. H. and Heathington, K. W. (1988). Motorist understanding of railroad-highway grade crossing traffic control devices and associated traffic laws. *Transportation Research Record: Journal of the Transportation Research Board*, (1160).

Roth, E. M., Malsch, N., and Multer, J. (2001). Understanding how train dispatchers manage and control trains: results of a cognitive task analysis. Technical report, United States. Federal Railroad Administration.

Şahin, İ. (1999). Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research Part B: Methodological*, 33(7):511–534.

Salido, M. A., Barber, F., and Ingolotti, L. (2008). Robustness in railway transportation scheduling. In *2008 7th World Congress on Intelligent Control and Automation*, pages 2880–2885. IEEE.

Saunders, C., Gammerman, A., and Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521.

Savage, I. (2006). Does public education improve rail–highway crossing safety? *Accident Analysis & Prevention*, 38(2):310–316.

Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.

Sehitoglu, T., Mussanov, D., and Dick, C. T. (2018). Operational schedule flexibility, train velocity and the performance reliability of single-track railways. In *Proceedings of the transportation research board 97th annual conference*.

Shih, M.-C., Dick, C. T., Sogin, S. L., and Barkan, C. P. (2014). Comparison of capacity expansion strategies for single-track railway lines with sparse sidings. *Transportation Research Record*, 2448(1):53–61.

Soderstrom, T. A., Himmelblau, D. M., and Edgar, T. F. (2001). A mixed integer optimization approach for simultaneous data reconciliation and identification of measurement bias. *Control Engineering Practice*, 9(8):869–876.

Surface Transportation Board (2016). Decision 45126: CSX transportation, inc.–acquisition of operating easement–grand trunk western railroad company.

Tjoa, I.-B. and Biegler, L. (1991). Simultaneous strategies for data reconciliation and gross error detection of nonlinear systems. *Computers & Chemical Engineering*, 15(10):679–690.

Tobias, A., House, D., and Wade, R. (2010). Planning capacity improvements in the chicago–milwaukee–madison rail corridor using the rail traffic controller (rtc) rail operations simulation model. In *Joint Rail Conference*, volume 49071, pages 297–305.

Tong, H. and Crowe, C. M. (1995). Detection of gross errors in data reconciliation by principal component analysis. *AIChE Journal*, 41(7):1712–1722.

Törnquist, J. (2006). Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. In *Proceedings of the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)*, volume 2.

Törnquist, J. (2007). Railway traffic disturbance management—an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. *Transportation Research Part A: Policy and Practice*, 41(3):249–266.

Törnquist, J. and Persson, J. A. (2007). N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological*, 41(3):342–362.

Tumulty, B. (2015). Google maps will highlight rail crossings. *USA Today*. 29 June 2015.

United States Department of Transportation and Federal Railroad Administration (2007). Intelligent transportation system/positive train control at highway-rail intersections. *Research Results*, (RR 07-20).

Vromans, M. J., Dekker, R., and Kroon, L. G. (2006). Reliability and heterogeneity of railway services. *European Journal of Operational Research*, 172(2):647–665.

Wang, P. and Goverde, R. M. (2016). Train trajectory optimization of opposite trains on single-track railway lines. In *Proceedings of the International Conference on Intelligent Rail Transportation (ICIRT)*, pages 23–31.

Wang, R. and Work, D. B. (2015). Data driven approaches for passenger train delay estimation. In *Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, pages 535–540.

Weatherford, B. A., Willis, H. H., Ortiz, D. S., Mariano, L. T., Froemel, J. E., and Daly, S. A. (2008). *The State of US Railroads: A Review of Capacity and Performance Data*. Rand Corporation.

Wronski, R. (2008). Time limits on trains at crossings barred. *Chicago Tribune*. 26 January 2008.

Wyman, O. (2016). Assessment of European railways: Characteristics and crew-related safety.

Yuan, J. and Hansen, I. A. (2007). Optimizing capacity utilization of stations by estimating knock-on train delays. *Transportation Research Part B: Methodological*, 41(2):202–217.

Zhao, M., Garrick, N., and Achenie, L. (1998). Data reconciliation based traffic count analysis system. *Transportation Research Record: Journal of the Transportation Research Board*, 1625:12–17.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.