Rail dispatch analysis methods using optimization and empirical data

William Barbour, Ph.D.
Vanderbilt University
February 19, 2021
Personal background

• B.S. Biosystems Engineering, University of Tennessee, Knoxville, 2015.

• M.S. Civil Engineering, University of Illinois at Urbana-Champaign, 2017.

• Network Modeling and Analytics Intern, CSX Transportation, 2017.

• Ph.D. Civil Engineering, Vanderbilt University, 2020.
History of rail “cyberphysical” systems

- Automatic block systems were one of the earliest forms of cyberphysical automation – 1872.
- Railroads were pioneers in computer systems and data processing.
  - Rolling stock tracking.
  - Field terminals at yards.
  - Computerized signaling.
  - Automatic invoicing.

[Images: Early centralized traffic control, “Semaphore” signaling]
Rail challenges today

A Self-Driving Freight Truck Just Drove Across the U.S. [Railway Age]

Railroads Losing Price Advantage Over Trucking

The dwindling pricing advantage means that there is less financial incentive for shippers to transport cargo "intermodal" rather than just utilizing trucking for the entire length of the trip.

Normalized cost per unit-mile, 2,000-mile sample

[Oliver Wyman, 2018]
Digital transformation in freight rail

Autonomous Trains Are Pouncing Rail

“Railways 2.0”: A Coming Bonanza?

Written by Alan Brody

Freight trains are our future

Excerpt: Power Trip

Michael E. Webber  May 9, 2019
Rail transformation challenges

- Railroads have efficiency advantage, but very costly in terms of infrastructure.
- Terminals and junctions are choke points for traffic.
- Traffic is heterogeneous (e.g., power, length, tonnage, cargo).
- Delay triggers are unavoidable (e.g., weather, other railroads, customers, mechanical).
- Even small delays can harm the ability to plan precisely.
Overarching research question: How do we develop data-driven methods for freight railroads to accelerate operational improvements?
Example workflow of data-driven railroading

Real-time data streams

Historical archive

Data cleaning/reconciliation

Real-time operation systems

Routing

Scheduling

Asset allocation

ETA prediction

Information consumers

Dispatch analysis
Example workflow of data-driven railroading

Real-time data streams → Historical archive → Data cleaning/reconciliation → ETA prediction

Real-time operation systems
- Routing
- Scheduling
- Asset allocation
- ... 

Information consumers
- Dispatch analysis

Automatic data reconciliation
- Automatically fix erroneous and missing rail dispatch data, guaranteeing feasibility of train trajectories.
- Used for any optimization-based dispatching model.
Example of erroneous rail data

- **Research problem:** perform data cleaning *automatically* on large-scale datasets, by fixing errors and missing points to guarantee feasibility in rail dispatch data.
- *For datasets of millions of points, manual data cleaning is impossible.*
Data reconciliation case study

• **Method**: minimize changes to data while satisfying feasibility.
• Optimization problem uses constraint set from optimal dispatching problem, repurposed for data feasibility.
• Apply on dataset with 75 million timing points.
• Error rate: < 0.5%; equivalent to 50,000+ errors/year!
• Reconciliation process is extremely fast: 1 million points/hour.
• Outperformed interpolation methods in timing and feasibility.

Example error found by data reconciliation

[ Barbour, et al., 2020 ]
Data-driven dispatch analysis

- Method to analyze train performance and dispatching relative to a schedule or optimal plan.
- Can look at both historical and real-time data.
Data-driven dispatch analysis

- **Research problem:** determine the actions of trains or dispatching that negatively impact the ability of the railroad to run to a schedule or plan.
- Isolating specific events or periods of time that are problematic can help attribute the sources of delays or deviations and adapt future plans.
Continual dispatching process

• Dispatching is often implemented in a model predictive control framework: initialize plan, begin executing, replan considering new state.
  – [Törnquist, 2006; Fang, 2015]

• Process of considering deviations from original plan and develop a new plan – “replanning”.
  – [Corman, 2012; Gestrelius, 2012; Pellegrini, 2016]

• When does the cost of the new plan increase? Which deviations caused the increase? And why?
Not all schedule deviations are created equal.

Long trains!
Need large siding track.
Motivating questions

**Question 1:** What is the overall (current + future) dispatching cost, given the current network state?

**Question 2:** How could the network state have been altered to improve replanning options (lower future cost)?

**Question 3:** Which trains have the largest impact on the ability to run to schedule?

[Barbour, et al., 2020]
Optimal dispatch

Reminder:

**Optimal dispatch problem**

**minimize:** total train runtime

**subject to:** speed constraints
meet constraints
pass constraints
follow constraints
length constraints

Reminder:

**Optimal dispatch problem**

**minimize:** \( f(x, z) \)

**subject to:** \( A_1 x + A_2 z \leq b \)

\( x \in \mathbb{R}^p_+, z \in \mathbb{Z}^q \)

where:

- \( x \) – continuous variables
  (time at which trains reached points on network)
- \( z \) – integer/binary variables
  (e.g., track assignment, meets or pass location, follow order, etc.)

[Petersen, 1986; Higgins, 1996; Wang, 2016; Bollapragada, 2018]
Example constraints in dispatch optimization model

- Must not exceed free run segment speed (minimum runtime).

\[
\text{IF } \sigma_{i,s} = 0, \text{ THEN } x_{i,m} \geq x_{i,m-1} + T_{i,m},
\]

where:
- \(\sigma_{i,s}\) – binary variable, =1 if train \(i\) took siding track \(s\).
- \(x_{i,m}\) – arrival time of train \(i\) at end of track \(m\).
- \(T_{i,m}\) – minimum free run travel time of train \(i\) on track \(m\).

- Trains taking siding must not exceed siding speed.

\[
\text{IF } \sigma_{i,s} = 1, \text{ THEN } x_{i,s} \geq x_{i,s-1} + U_{i,s}
\]

where:
- \(x_{i,s}\) – arrival time of train \(i\) at end of siding track \(s\).
- \(U_{i,s}\) – minimum travel time of train \(i\) on siding track \(s\).

Note: written here logically but coded as mixed integer constraint.
Example constraints in dispatch optimization model

Additional constraints:
- Meet events on siding tracks.
- Overtake events on siding tracks.
- Follow headway time.
- Train vs. siding length
- Compound overtake events.
- Simultaneous meet/overtake events.
- Initial condition (time and location).
- Final condition (location).

Approximate problem size for 20 trains (12 hours):
- 6,000 variables
  - 1,000 continuous
  - 4,000 binary
  - 1,000 integer
- 16,000 constraints
  - 13,000 inequality
  - 3,000 equality
Case study dataset

• S&NA North CSX subdivision between Nashville, TN, and Birmingham, AL.
  – 190 miles.
  – 17 sidings.
  – 15-25 trains per day.
  – 4-hour free-run traversal time.
• Determine model parameters from 2 years of data (2014-15):
  – Main line segment times.
  – Siding segment times.
  – Meet headways.
  – Train follow headways.
• Apply methods to data collected over 1 month of operation.
What is the overall (current + future) dispatch cost, given the current network state?

• Fix network state up to time $\tau$.
• Minimize runtime of all trains, *replanning* from the network state at $\tau$.
• Replan is best possible future performance (after time $\tau$).
Dispatch replanning process

\[ \tau = 300 \]

time (minutes)

position on track

Optimal
Siding / Dbl. Track
Dispatch replanning process

\[ \tau = 300 \]

- Empirical
- Optimal
- Siding / Dbl. Track

position on track

time (minutes)
Dispatch replanning process

\[ \tau = 300 \]

- **Empirical**
- **Replan after \( \tau = 300 \)**
- **Optimal**
- **Siding / Dbl. Track**
Replanning cost analysis

- When $\tau = 0$, replan is same as baseline optimal plan.
- When $\tau = 600$, nothing remains for replanning.
- Evaluate the cost of the replan at time $\tau$, assuming fixed network state at that time and optimal replanning moving forward.
Replanning cost analysis

- When $\tau = 0$, replan is same as baseline optimal plan.
- When $\tau = 600$, nothing remains for replanning.
- Evaluate the cost of the replan at time $\tau$, assuming fixed network state at that time and optimal replanning moving forward.

Cost of replan = cost of state up to $\tau +$ cost of new plan after $\tau$
What is the overall (current + future) dispatch cost, given the current network state?

- Fix network state up to time $\tau$.
- Minimize runtime of all trains, *replanning* from the network state at $\tau$.
- Replan is best possible future performance (after time $\tau$).

**Dispatch trains minimizing total runtime...**

...subject to normal dispatch constraints...

...and network state (train trajectories) up to $\tau$ fixed to observed values.

**minimize:**

$$\sum_{i \in I} x_{i,q_i} + \sum_{j \in J} x_{j,q_j}$$

**subject to:**

$$A_1 x + A_2 z \leq b$$

$$\|x_{\tau^-} - \tilde{x}_{\tau^-}\|_1 = 0$$

- Completion times of train $i$ and train $j$.
- Vector of empirical timing points before time $\tau$.
- Vector of corresponding decision vars.
\( \tau = 0 \)

- Trains started: 0
- Trains finished: 0
- Trains remaining: 17
- Trains running: 0
Trains started: 2
Trains finished: 0
Trains waiting: 15
Trains running: 2
Trains started: 2
Trains finished: 0
Trains waiting: 15
Trains running: 2

\[ \tau = 60 \]
\( \tau = 90 \)

- Trains started: 4
- Trains finished: 0
- Trains waiting: 13
- Trains running: 4
Trains started: 4
Trains finished: 0
Trains waiting: 13
Trains running: 4

$\tau = 120$
Trains started: 7
Trains finished: 1
Trains waiting: 10
Trains running: 6

\[ \tau = 150 \]
Trains started: 8
Trains finished: 1
Trains waiting: 9
Trains running: 7

\[ \tau = 180 \]
Trains started: 8
Trains finished: 2
Trains waiting: 9
Trains running: 7

\[ \tau = 210 \]
Trains started: 9
Trains finished: 4
Trains waiting: 8
Trains running: 5

\[ \tau = 240 \]
Trains started: 11
Trains finished: 4
Trains waiting: 6
Trains running: 7
Trains started: 12
Trains finished: 4
Trains waiting: 5
Trains running: 8
Trains started: 14
Trains finished: 4
Trains waiting: 3
Trains running: 10
Trains started: 15
Trains finished: 4
Trains waiting: 2
Trains running: 11

\[ \tau = 360 \]
Trains started: 15
Trains finished: 4
Trains waiting: 2
Trains running: 11

\[ \tau = 390 \]

Diagram showing train schedules and runtime data. The inset graph illustrates the total train runtime (minutes) against \( \tau \) (minutes), with different lines representing empirical, baseline, and replanned scenarios.
Trains started: 15
Trains finished: 4
Trains waiting: 2
Trains running: 11

\[ \tau = 420 \]
Trains started: 15
Trains finished: 5
Trains waiting: 2
Trains running: 10

\[ \tau = 450 \]
Trains started: 16
Trains finished: 6
Trains waiting: 1
Trains running: 10

$\tau = 450$
Trains started: 17
Trains finished: 6
Trains waiting: 0
Trains running: 11

$\tau = 450$
Trains started: 17
Trains finished: 17
Trains waiting: 0
Trains running: 0

\( \tau = 450 \)
Comparison of cost by day

• Find varied trends in evolution of replanned lower bound.
• Sharp jumps in lower bound indicate problematic decisions or consequential events.
• Depends on distribution of trains within dispatch time window.
How could the network state have been altered to improve replanning options?

- Baseline cost = $r_0$.
- Replanning cost at $\tau = r_\tau$.
- Require reduction in replan cost: $r_0 \leq r' \leq r_\tau$.

- Achieved by making network state alterations up to $\tau$.
- Network state alterations are changes to a train’s segment runtime.
Formulation of altered decision making

Minimize alterations to network state before \( \tau \)...

...subject to normal dispatch constraints...

...and require reduce replanned dispatch cost \( r' \).

\[
\text{minimize:} \quad \sum_{i \in I} \sum_{n=p_i+1}^{q_i} |(x_{i,n} - x_{i,n-1}) - (\tilde{x}_{i,n} - \tilde{x}_{i,n-1})| + \sum_{j \in J} \sum_{n=p_j-1}^{q_i} |(x_{j,n} - x_{j,n+1}) - (\tilde{x}_{j,n} - \tilde{x}_{j,n+1})|
\]

subject to:

\[
A_1 x + A_2 z \leq b
\]

\[
\sum_{i \in I} x_i q_i + \sum_{j \in J} x_j q_j \leq r'
\]

Completion time of train \( i \) and train \( j \).
Dispatch replanning process

\[ \tau = 300 \]

- **Empirical**
- **Replan after \( \tau = 300 \)**
- **Optimal**
- **Siding / Dbl. Track**

The diagram shows the position on track over time (minutes) with different replanning processes marked at \( \tau = 300 \) minutes.
Question 2 results

- Zero reduction requires zero alteration.
- Reduction to baseline optimal requires complete alteration.
- Looking for small network state alterations that have a large impact on replan cost.
Question 2 results

- Zero reduction requires zero alteration.
- Reduction to baseline optimal requires complete alteration.
- Looking for small network state alterations that have a large impact on replan cost.
- Reducing lower bound replan runtime by 188 minutes can be achieved by making a 45-minute adjustment to empirical dispatch: a 4:1 effect.
Example of impactful alteration

Alterating these three points before $\tau$ by 20 minutes…

…allowed future runtime reduction of additional 25 minutes.
Question 3: impact of individual trains

Which trains have the largest impact on the ability to run to schedule?

- Fix trajectory of one train, $w$, to its empirical values.
- Measure impact on the trajectories of other trains versus their baseline optimal values.

- Total planned runtime = 120 min.
- Total replanned runtime = 210 min.
- Train 1 primary effect = +40 min.
- Train 1 secondary effect = +50 min.
Formulation of individual train impact

Dispatch trains minimizing total runtime...

...subject to normal dispatch constraints...

...but with one train fixed to its empirical trajectory.

\[
\text{minimize: } \sum_{i \in I} x_i,q_i + \sum_{j \in J} x_j,q_j
\]

subject to:

\[
A_1 x + A_2 z \leq b
\]

\[
\| x_w - \tilde{x}_w \|_1 = 0
\]

Completion time of train \( i \) and train \( j \).

Empirical timing points of train \( w \).

Train \( w \) timing decision vars.
Differences in train impact

- **Primary effect:** train’s own difference from optimal plan.
- **Secondary effect:** difference from optimal plan of other trains.
- Either effect can be negative but net effect must be runtime increasing, compared to baseline optimal plan.

![Graph showing added runtime for different trains, with primary and secondary added runtime indicated.]

**Small deviation from schedule, but large impact on other trains.**

**Large deviation from schedule, small impact on other trains.**
Example of particularly impactful train
Example of particularly impactful train

+11 minutes

R59

G28

Q11

Siding / Dbl. Track

Optimal

R59 fixed

Secondary effect

0 50 100 150 200

position on track

time (minutes)
Example of particularly impactful train
• Assess trend in primary/secondary train impact across a month.

• Most trains had a 1.4-minute secondary effect for every minute of primary effect.

• Numerous exceeded a 2.0 minute/minute effect.
Conclusions

- **Data reconciliation** is a more effective means of fixing erroneous rail dispatch data than manual inspection or interpolating missing points.
- **Dispatch analysis** can reveal time periods, events, and trains that were critical to dispatch quality and performance. Analyzing these elements further can reveal productive improvements or new strategies.
Rail dispatch analysis methods using optimization and empirical data

William Barbour, Ph.D.
Vanderbilt University
February 19, 2021
S&NA North siding runtimes

- Significant disparity in runtimes between pairs of meeting trains at sidings.
- Most sidings are short relative to trains (i.e., 1.5-2x).
Dispatch modification results

- Examine required empirical alteration across values of $\tau$.
- Overall consistent.
- Some points in time where alteration effect is especially high (e.g., 180 minutes).
• Meeting trains: whichever arrives first must clear with safety headway.

\[
\text{IF } \pi_{i,j,m} = 1, \text{ THEN } x_{i,m} + H_{i,j} \leq x_{j,m+1} \quad \text{ELSE } x_{j,m} + H_{i,j} \leq x_{i,m-1}
\]

where:

- \( \pi_{i,j,m} \) – binary; meet occurred between trains \( i \) and \( j \) on track \( m \).
- \( H_{i,j} \) – minimum safety headway between trains \( i \) and \( j \).

• Overtaking trains: following train must follow with safety headway.

\[
\text{IF } \phi_{i_1,i_2,m} = 1, \text{ THEN } x_{i_1,m} + H_{i_1,i_2} \leq x_{i_2,m}
\]

where:

- \( \phi_{i_1,i_2,m} \) – binary; pass occurred between trains \( i_1 \) and \( i_2 \) on track \( m \).
- \( H_{i_1,i_2} \) – minimum safety headway between trains \( i_1 \) and \( i_2 \).
Optimal dispatch formulation

Reminder:

Data reconciliation problem

\[
\text{minimize: } \quad \text{changes to data} \\
\text{subject to: } \quad \text{speed constraints} \\
\quad \text{meet constraints} \\
\quad \text{pass constraints} \\
\quad \text{follow constraints} \\
\quad \text{length constraints}
\]

Minimize changes to empirical data that make it feasible.

Reminder:

Data reconciliation problem

\[
\text{minimize: } \sum_{x,z} \|x_\Omega - \tilde{x}_\Omega\|_1 + \|x_\Psi - x_{\text{des}}\|_1 \\
\text{subject to: } \quad A_1 x + A_2 z \leq b, \\
\quad x \in \mathbb{R}_+^p, z \in \mathbb{Z}^q
\]

where:

\(x\) – continuous variables (train timing points)
\(z\) – integer variables (e.g., meets/passes)

Minimize difference between feasible trajectories (\(x\)) and empirical data.
Primer: optimal dispatching

• Finding best feasible schedule for given train lineup.
  – “Best” usually in terms of delay/runtime minimization.
  – “Feasible” means that schedule and trajectories satisfy problem constraints (physical, logical, and self-imposed).
  – Many problems fit into this abstracted form (e.g., Petersen et al. 1986, Higgins et al. 1996, Gestrelius et al. 2017, Lamorgese et al. 2017, …).

Optimal dispatch problem

\[
\begin{align*}
\text{minimize:} & \quad f(x, z) \\
\text{subject to:} & \quad A_1 x + A_2 z \leq b \\
\text{where:} & \quad x \in \mathbb{R}^p, z \in \mathbb{Z}^q
\end{align*}
\]

Measure of train delay/runtime minimization, possibly weighted.

Mixed integer linear constraint set encoding dispatching logic and operational rules.

\(x\) – continuous variables (train timing points)
\(z\) – integer variables (e.g., meets/passes)
Data reconciliation formulation

- Process of automatically fixing data errors and imputing missing values is called *data reconciliation*.
- Fixes errors using smallest possible correction to reach feasibility.
- Let $\Omega$ denote existing data, and $\Psi$ denote missing data.

Minimize $L_1$ difference between reconciled $(x_\Omega)$ and historical $(\tilde{x}_\Omega)$ when not missing.

For missing, data impute points as close ($L_1$) as possible to desired runtime, $x^{\text{des}}$.

Reconciled data obeys same mixed integer linear constraint set as optimal dispatch problem.
Data reconciliation formulation

- Use $L_1$ penalty on deviations from historical data for continuous variables (timing points).

\[
g(x_\Omega - \tilde{x}_\Omega, z_\Omega - \tilde{z}_\Omega) = \|x_\Omega - \tilde{x}_\Omega\|_1
\]

- Imputed data regularizes train speed to $x^{\text{des}}$ (e.g., mean).

\[
h(x_\Psi, z_\Psi) = \|x_\Psi - x^{\text{des}}\|_1
\]

- Objective function becomes:

\[
\text{minimize: } \|x_\Omega - \tilde{x}_\Omega\|_1 + \|x_\Psi - x^{\text{des}}\|_1
\]
## Reconciliation objective functions

<table>
<thead>
<tr>
<th>Objective Function</th>
<th>Equation</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_1 ), constant speed</td>
<td>( |x_\Omega - \tilde{x}_\Omega|<em>1 + |x</em>\Psi - x^{cs}|_1 )</td>
<td>( x^{cs} ) is interpolated assuming constant speed across track segments with missing data. ( x^{cs} ) – see previous.</td>
</tr>
<tr>
<td>( L_2 ), constant speed</td>
<td>( |x_\Omega - \tilde{x}_\Omega|<em>1 + |x</em>\Psi - x^{cs}|_2 )</td>
<td>( x^{cs} ) – see previous.</td>
</tr>
<tr>
<td>( L_1 ), average segment speed</td>
<td>( |x_\Omega - \tilde{x}_\Omega|<em>1 + |x</em>\Psi - x^{ss}|_1 )</td>
<td>( x^{ss} ) is interpolated assuming track segment travel times are distributed proportional to average historical segment speeds. ( x^{ss} ) – see previous.</td>
</tr>
<tr>
<td>( L_2 ), average segment speed</td>
<td>( |x_\Omega - \tilde{x}_\Omega|<em>2 + |x</em>\Psi - x^{ss}|_2 )</td>
<td>( x^{ss} ) – see previous.</td>
</tr>
<tr>
<td>( L_1 ), train type average segment speed</td>
<td>( |x_\Omega - \tilde{x}_\Omega|<em>1 + |x</em>\Psi - x^{tt}|_1 )</td>
<td>( x^{tt} ) is interpolated assuming track segment travel times are distributed proportional to average historical segments speeds, grouped by corresponding train type. ( x^{tt} ) – see previous.</td>
</tr>
<tr>
<td>( L_2 ), train type average segment speed</td>
<td>( |x_\Omega - \tilde{x}_\Omega|<em>2 + |x</em>\Psi - x^{tt}|_2 )</td>
<td>( x^{tt} ) – see previous.</td>
</tr>
</tbody>
</table>
Examples of erroneous rail data

- Data fusion and aggregation from dispatch system introduces some errors in trajectory points.
- Errors manifest in violations of operational constraints (e.g., headway violation) and meet/pass logic.
Test 1: fixing historical dataset

- Tested 6-month dataset from CSX subdivision.
- Manually inspected 1 week of data to validate.
- An error was found on average every 4 hours of data.
Test 2: synthetically-decimated dataset

• Deleted varying numbers of *known* data points around meet and pass events, total of 2-6 points.

• Impute data point using:
  – Linear interpolation
  – Data reconciliation

• Assess imputed data:
  – Timing error of imputed data points.
  – Fraction of meets/passes that occur at a *feasible* location (i.e., siding).
  – Fraction of meets/passes that occur at the *correct* location (i.e., indicated by known data).
Test 2: reconciliation results

- Created synthetically-decimated dataset by deleting varying numbers of *known* data points around meet and pass events and imputing by linear interpolation and reconciliation.

- Reconciliation reduced timing error of imputed points by 10%.

- Reconciliation found feasible locations for meet/pass events in 100% of cases.

- Correct location found by reconciliation in 90+% of cases; 20-50% with interpolation.

*Barbour, Kuppa, Work. 2019*
Test 3: comparison of objective functions

- Six objective functions tested on ability to reconstruct synthetically-decimated dataset with minimal timing error.
  - $L_1$ vs. $L_2$ norm.
  - Interpolation by constant speed vs. average segment speed vs. train type average segment speed.
- Found large MSE benefit with $L_2$ objective functions, at slight expense to MAE.
- Average segment speed reduced error $\sim 10\%$ over constant speed.