Testing Intermodal Terminal Optimization Models With Simulation

Patricia Randall, PhD

October 20, 2019

PRINCETON CONSULTANTS
Information Technology and Management Consulting

www.princeton.com

Agenda

- Parking Optimization Overview
- Testing with Simulation
- Optimization Approaches
- Lessons Learned

Agenda

- Parking Optimization Overview
- Testing with Simulation
- Optimization Approaches
- Lessons Learned

Objectives

Develop

Model to optimize parking assignments for equipment in an Intermodal terminal

Test

Using a simulation of Intermodal terminals

- Determine solution quality
- Identify bugs and design issues in model
- Refine model solution based on testing

Parking Policies in Intermodal Terminals

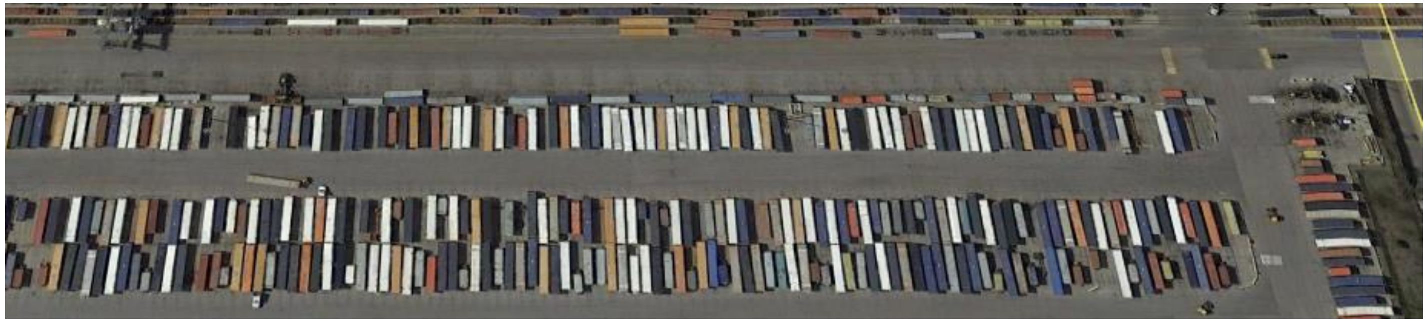
Hundreds of containers and trailers will be parked every day in an Intermodal terminal yard

Where each unit is parked will impact train unloading and loading times as well as truck dwell time

Parking also affects general terminal congestion and terminal safety for both the outside truckers and the hostlers who work on the terminal

Parking Policies in Intermodal Terminals

In wheeled terminals, containers on chassis and trailers are parked in spaces on the terminal lot

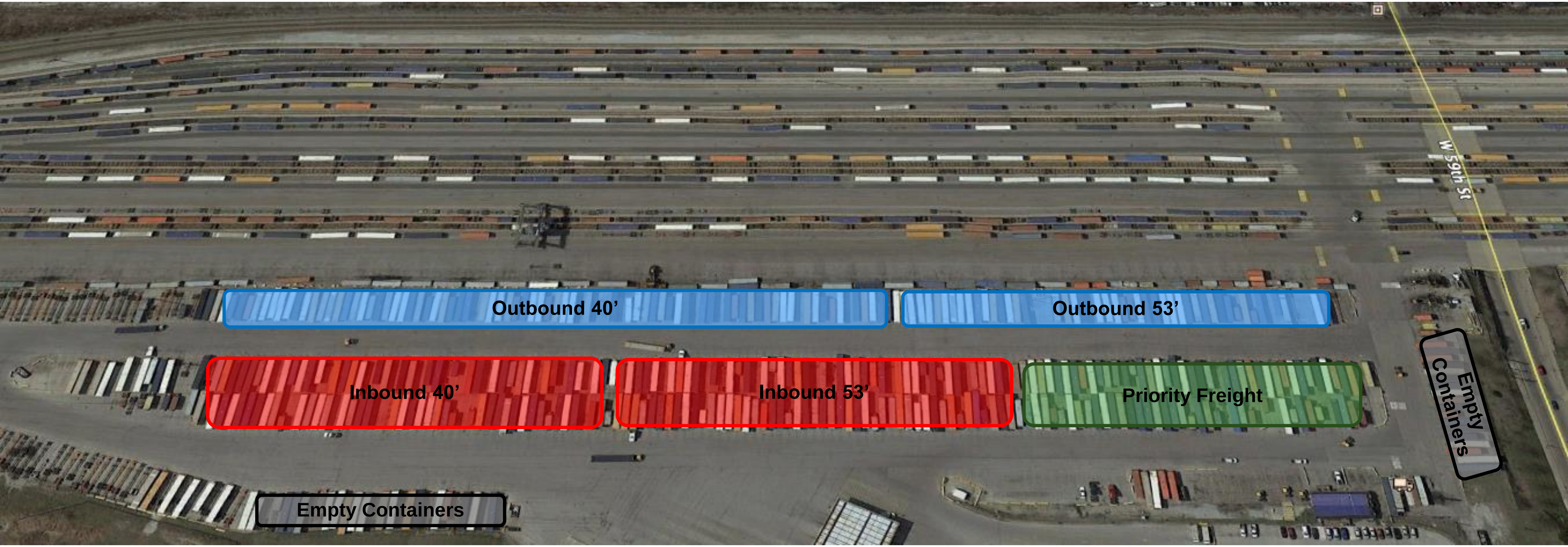


Parking policies are often based on the freight's attributes

- Inbound (heading out the outgate) or outbound (leaving on a train)
- Length – 20' vs. 40' vs. 53'
- Group together freight for the same train
- Priority customers or priority freight (e.g. expensive commodity)
- Empty vs. loaded

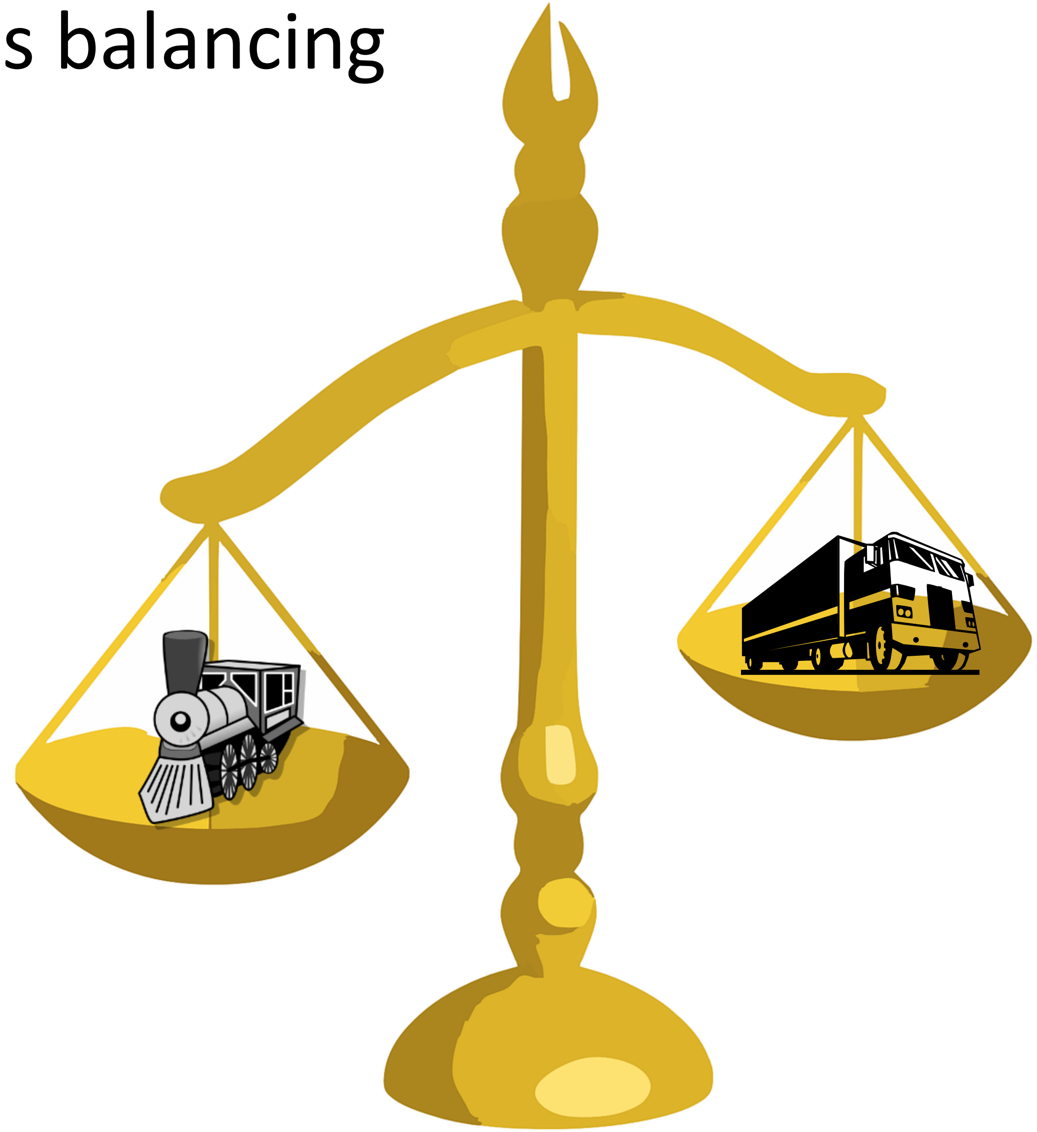
Parking Policies in Intermodal Terminals

For instance, a terminal might split up the spaces like this:



Parking Policy Optimization

- Optimizing the parking policy means balancing multiple objectives
  - Minimize physical travel time/miles
  - Minimize loading and grounding time
  - Maximize compliance to terminal parking preferences
  - Minimize outside trucker proximity to tracks



Agenda

- Parking Optimization Overview
- Testing with Simulation
- Optimization vs. Reinforcement Learning
- Lessons Learned

Testing Your Model

Regardless of the approach and algorithm you use, you need a way to test your model's results before you roll it out to the terminals

Critical for

- Validating data quality and completeness
- Ironing out interface design kinks
- Fine tuning the model
- Gaining end user buy-in

Agent-Based Simulation

- Representing a complex system by a collection of agents that follow defined behavior rules
  - Agents mimic the behavior of their real-world counterparts
  - Agents have goals and behaviors and control themselves based on them
  - Agents are capable of making autonomous decisions
  - Behavior rules can be as simple or complex as necessary

❖ **Good for modeling individual entities with independent behavior**

❖ **Can explicitly model complex interactions between agents to better mimic the real world**

❖ **Aligns well with object-oriented programming**

When to use Agent-Based Simulation

When the entities in the problem can be naturally represented as agents

When entities have relationships or interactions with other entities, especially dynamic ones

When entities need to learn or adapt their behavior to their environment or other entities' behavior

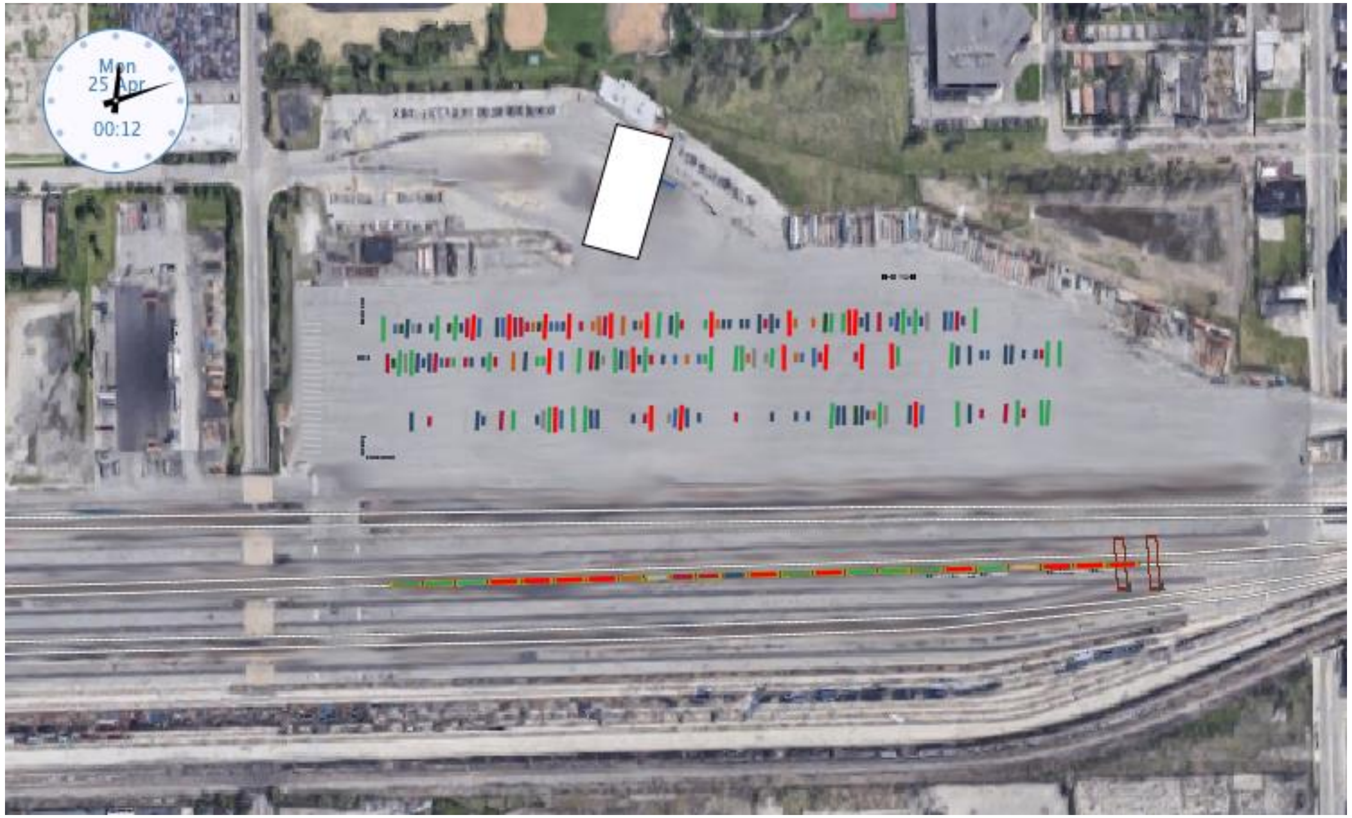
When entities have spatial awareness, like when moving on a map

When past experiences are not enough to predict future emerging scenarios

When decisions made within the simulation need to be a result of the entities, rather than an input to the simulation

Simulation Development and Baselining

- Simulation was developed for the chosen pilot terminal and baselined against actual operations to validate that it was accurately modeling the terminal
  - Compare key metrics output from the simulation with real-world values
  - SMEs watched the simulation's animation to spot any major issues with resource behavior



Agenda

- Parking Optimization Overview
- Testing with Simulation
- Optimization Approaches
- Lessons Learned

Parking Policy Optimization – MIP Approach

- Our first approach was a traditional mixed integer programming model (MIP) to provide the optimal assignments
- Simulation provided the first heavy load tests as well as tests with a variety of parking assignment requests
  - Allowed the modelers to test different "What If" scenarios very quickly with hundreds of assignment tests per run
- But we also found performance was an issue – the MIP was too slow to run real-time for gate operations
  - So we tried a second approach – Reinforcement Learning

What is Reinforcement Learning?

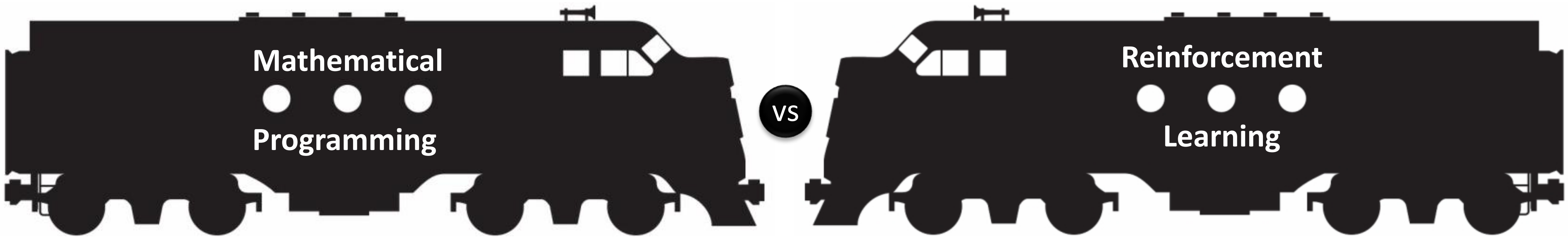
- A type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences

Supervised learning — You’re feeding labeled data into a neural network. Essentially, you’re telling the machine what it’s seeing.

Unsupervised learning — You’re not providing labeled data. You have no specified goal or target; the machine uncovers structural aspects of the data as it learns.

Reinforcement learning –You’re providing partial information to the model, but not providing a correct answer or action. Reinforcement learning is a form of supervised learning.

MIP vs. Reinforcement Learning



	Math Programming	Reinforcement Learning
Technology	Mature, proven technology	Cutting-edge technology that is still evolving
Familiarity within OR community	Well understood by OR practitioners	Most OR practitioners are probably not very familiar with RL
Solution Optimality	Guaranteed optimality (within gap)	Not guaranteed an optimal solution but probably can get one that is good enough
Performance for real-time gate operations	Performance could be a challenge for use in real-time gate operations	Training is time-consuming but once trained, utilizing the policy is very fast

Training Reinforcement Learning with Simulation

Why train Reinforcement Learning algorithms with simulations?

Risk-free space for trying new ideas

Unlimited amounts of relevant, clean, structured, labeled data

Can generate data from systems not seen in real life (e.g. adding RTGs to an existing intermodal facility)

Can replay historical data to provide the “what” without knowing the “why”

Parking Policy Optimization – RL Approach

- The Reinforcement Learning algorithm performed as well as the MIP model for simple reward functions (i.e. objectives)

- It takes longer to train the algorithm and create the policy

  ….But then using the policy is faster than solving the MIP

- Next Steps: Use the MIP to calculate the reward function and compare the Reinforcement Learning policy vs. the MIP for more complex rewards

Agenda

- Parking Optimization Overview
- Testing with Simulation
- Optimization Approaches
- Lessons Learned

Some Lessons Learned

Choose the right type of simulation for your problem: agent-based, discrete event, or a hybrid

Test your model early and often using a simulation before rolling out to the Field

Choose simulation software that meets your key criteria like customization or ease of use

Validate data quality and completeness as an initial step before looking at model results

Reinforcement Learning

Simplicity is key for both reward function and observation state

Scenario variability is important to ensure the algorithm doesn't tune itself to a subset of states

Thank You and Questions

PRINCETON CONSULTANTS
Information Technology and Management Consulting

www.princeton.com

Patricia Randall
Director
PRandall@princeton.com
www.princeton.com
609-987-8787