Inverse Reinforcement Learning for Predicting Ship Sequencing and Time Spent in Port Container Terminals

Nikolay Aristov, Ziyan Li, Elenna R. Dugundji

naristov@mit.edu, ziyan810@mit.edu, elenna_d@mit.edu

Massachusetts Institute of Technology, Center for Transportation and Logistics, Cambridge, USA

Keywords: Inverse Reinforcement Learning, Sequencing, Container Ports

1 Introduction

Ports are major entry points for imported goods into the U.S. Every US dollar of trade flowing through a port will directly or indirectly generate an additional US $4 in global industry output [7]. Effective management of goods flow requires clear visibility into logistics, congestion, port sequencing, and terminal scheduling. The Freight Logistics Optimization Works (FLOW), an initiative launched by U.S. Department of Transportation (U.S. DOT), seeks to enhance this visibility and improve cooperation among key supply chain stakeholders.

Traditionally, berth scheduling models assume a known objective function, such as minimizing completion time [4] or maximizing the number of ships served [2].

In our study, we assume that each terminal operates with its own expert strategy for ship sequencing. Given the impracticality of learning these strategies for every port, we sought to derive them from historical data. To achieve this, we studied Automatic Identification System (AIS) data, which collects vessel information throughout maritime voyages via radio frequency. We processed this data to model port operations and to train and test our models. We also applied Inverse Reinforcement Learning (IRL) algorithm [5] to learn the reward function that mimics ship's sequencing behavior of port authorities. Our approach builds on the work of [8], adapting it to the context of maritime logistics.

For this algorithm, we proposed two different environments and discuss their application. Analyzed ports and terminals are listed in Table 1. We also discussed results and proposed additional improvements of models and algorithms for future research.

Table 1. List of Ports and Terminals

Port	Terminal
New York / New Jersey	APM Terminals
New York / New Jersey	Maher Terminals

2 Methodology

In this section, we describe our approach for building ship trajectories at terminals, applying IRL algorithms for deriving reward function and measuring accuracy of predictions.

2.1 Building Trajectories

We defined a trajectory as a set of states and actions over a set of defined contiguous timestamps. Each step in a trajectory consists of a tuple: (timestamp, current state, actions, next state), where

Timestamp: A predefined time window (6 hours in our case) during which ships are assumed to maintain the same position (e.g., staying at berth, staying at wait zone, etc.). We assume that re-positioning of ships between states happens immediately between timestamps.

Current state: The set of ships at specific positions at the terminal at the current timestamp. The list of positions is defined in Table 2.

Actions: The set of actions that transit ships from the current state into the next state. The list of actions is provided in Table 3.

Next state: The set of ships at particular positions at the terminal after actions are applied, forming the current state of the next trajectory step.

Table 2. List of Positions

Position	**Description**
Berths	Positions of ship's berthing
Waiting Area	Waiting Area near the port
Incoming	Positions to model incoming ships into the port, one position per time window up to 7 days

Table 3. List of Actions

Action	**Description**
Stay at Berths X	Ships stays at berth X, no position change
Leave the system	Ships leaves the system (after berthing)
Stay at waiting zone	Ship stays at waiting zone
Go to berth X	Ship goes to berth X from the current position (from waiting zone or closest to port incoming)
Go to waiting zone	Ship goes to waiting zone from the current position (from the closest to port incoming)
Come closer to incoming X -1	Ship goes closer one step to port to the next incoming zone. It is impossible for ship stay more then one timestamp at the same incoming zone and it is impossible for ship to return to previous incoming state

We applied two different methods to describe states in trajectories. **Model 1** (ship-based model) maps each ship to a position with the state length equal to the maximum number of ships in the system among all timestamps. Ships are sorted by their International Maritime Organization (IMO), and any missing states are set to 0. While this method provides a diversity to training data, it may reduce predictability.

Model 2 (state-based model) uses fixed position for states - one for each berth, and a set of position for waiting areas and incoming states. The number of waiting and incoming states is defined by the maximum number of ships at these states over all timestamps. Missing states are set to zero. This method could enhance predictability, especially at berths, but still may face challenges when multiple ships are present in waiting zones.

Apart from actions and states, we incorporate various features into our model: time features, ship size (grouped by clustering algorithm and one-hot encoded (OHE)), operators (OHE) and source-destination pairs (OHE). We use the entire dataset from 02/01/2015 to 09/30/2023 to derive the reward function, assuming that data during the COVID period will provide valuable insights into how ships were prioritized at the terminal.

2.2 Inverse Reinforcement Learning

To learn the expert priorities from the historical data, we implemented Inverse Reinforcement Learning (IRL) algorithms with a Maximum entropy framework [8]. This algorithm allows us to derive a reward function $R(s, a)$ for the environment, where a is the action taken in state s. This derived reward function then in turn provides predicted actions for ships from a predefined action set.

We model the berth scheduling problem as a Markov Decision Process (MDP), with the goal to maximize the expected value $V(s)$ of the observed expert trajectories, as described in Section 2.1. IRL with maximum entropy effectively handles the inherent ambiguity in deriving a reward function from observed behavior.

2.3 Measuring Accuracy

The primary objective of our model was to predict the time a ship would spend at the port. To evaluate the accuracy of our model, we predicted the next 1 to 10 states from the current state over a sequence of steps. This analysis allowed us to determine the maximum number of steps we could accurately predict without a significant decline in quality.

Additionally, we analyzed the prediction of ship movements within the terminal, focusing specifically on transitions to the berth, the waiting zone, and the duration spent in these states. Predicting the duration spent by the ship inside the port is our main task, as it will allow predicting congestion of future ports of call for this particular ship.

3 Results

In this section, we present our current findings on the accuracy of predicting berthing schedule, the time ships spend at berths and waiting areas, and berth occupations. These results provide insights into the model's performance in capturing key aspects of port operations.

3.1 Accuracy of Predicting Ship Positions

We calculated accuracy as a number of accurate prediction for the ship assigned or stayed at any berth and assigned or stayed at waiting zone compared to expert.

For comparison, we trained the model with different number of iterations, starting from 1 and then for every 10 iterations (10, 20...100). We also used randomly picked 14 initial states and evaluate based on actions next 10 steps. However, we didn't put any new ships in the calculation of states, but that's should not affect our prediction as we used four states per day and 7 days of incoming ships.

We compared the ship-based model behavior for two terminals (1) and from the results we could see that the best number of iterations are 40 and 90 for both terminals, but still the model has low prediction value for berths and wait zone occupation.

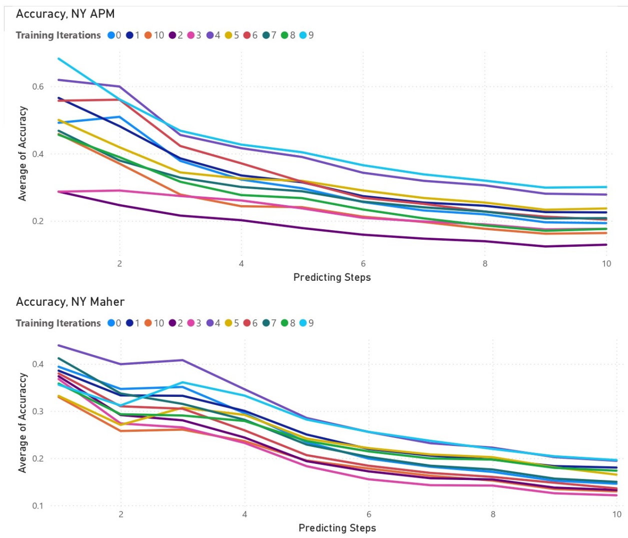

Figure 1. Accuracy of ship-based model for APM and Maher terminals

Comparing state-based model to ship-based model (2) for Maher terminal shows that state - based model with 90 iterations on the training is more reliable. However, state-based model requires more time for one training iteration due to increased number of states to process: while for ship-based model we are processing only allowed actions for ships at particular state, for state-based approach we have to iterate through all states at every state.

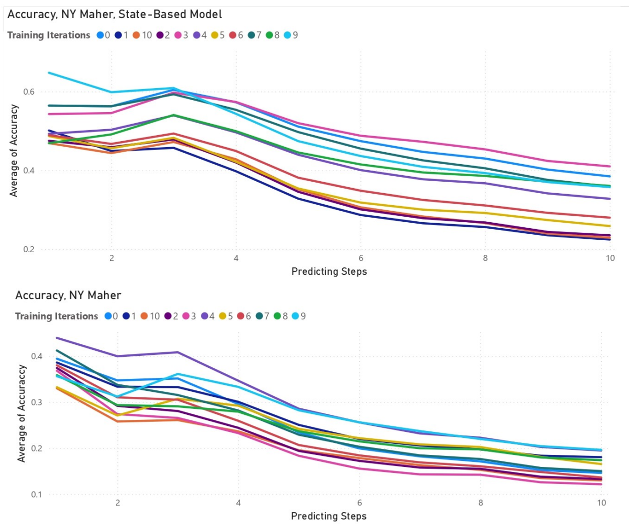

Figure 2. Accuracy of state-based and ship-based models for Maher terminal

3.2 Prediction of Time Spent

We also measured the performance of the Inverse Reinforcement Learning (IRL) models by comparing the predicted time spent at berth by ships with the actual data for both terminals. Table 4 summarizes the results, showing the total number of ships that spent more time, less time, and those with significant differences in time spent at berth.

Table 4. Spent time prediction

Terminal	State Based	Total Ships	Spent More	Spent Less	more 18h	more 1 day	more 1 day %
APM	No	71	3	60	25	15	21%
APM	Yes	73	1	59	27	16	22%
Maher	No	64	1	53	33	24	38%
Maher	Yes	65	14	41	24	17	26%

4 Conclusion

In this research, we introduce an IRL approach to derive the reward function of port authorities from ships' historical behaviors, especially berthing sequence and time spent at berths and waiting areas in port container terminals.

We compared two different environment models and their predictability based on two terminals of New York / New Jersey port: APM and Maher. For the accuracy prediction it is clear, that state-based model performs significantly better. We also observed, that 30 iterations of training performed slightly better on a long horizon for state-based model, than 90 iterations, that might be a good sign as training of state-based model consumes more time per iteration, than ship-based model.

However, further work is necessary to refine and validate our results. Specifically, we must compare the current models against a baseline to establish their relative performance and conduct a detailed analysis of feature contributions. Moreover, efforts should be focused on improving both the quality and speed of model training to enhance overall performance.

Additionally, exploring alternative IRL models, such as Linear Programming IRL [5], Feature Matching IRL [1], Bayesian IRL [6], or applications of IRL dealing with non-stationary nature [3].

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[2] A. Al-Refaie and H. Abedalqader. Optimal berth scheduling and sequencing under unexpected events. *Journal of the Operational Research Society*, 73(2):430–444, 2022.

[3] A. Likmeta, A. M. Metelli, G. Ramponi, A. Tirinzoni, M. Giuliani, and M. Restelli. Dealing with multiple experts and non-stationarity in inverse reinforcement learning: an application to real-life problems. *Machine Learning*, 110:2541–2576, 2021.

[4] B. Liu, Z.-C. Li, Y. Wang, and D. Sheng. Short-term berth planning and ship scheduling for a busy seaport with channel restrictions. *Transportation Research Part E: Logistics and Transportation Review*, 154:102467, 2021.

[5] A. Y. Ng, S. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[6] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

[7] J. Verschuur, E. Koks, and J. Hall. Ports' criticality in international trade and global supply-chains. *Nature communications*, 13(1):4351, 2022.

[8] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.