ED, DAKARAI, AND NATHAN

THE KRAKEN
THE PROBLEM

- How can we automate the scheduling of the Port of LA’s supply chain distribution?
- How can we maximize the number of cargo containers transported per day?
- How can we optimize the amount of time a trucker spends between importing and exporting goods?
MARINE TERMINAL TRAFFIC OPTIMIZATION

- Solutions exist!
  - i.e. vehicle routing problem and traveling salesmen
  - Utilized an extensive custom library for our purposes
  - Modified several target libraries to cater the design toward the seven criteria
MODEL ASSUMPTIONS

› All drivers arrive 24 hours after cargo has finished processing

› Any delays for a given driver do not disrupt performance of the algorithm

› Once a driver returns to the MTO queue, algorithm is recomputed
MODEL OVERVIEW

- Fleet of $X$ trucks
  - Pick $Y$ trucks to route, and iterate through all $X$.
  - Goal is to maximize efficiency by minimizing the time truckers spend entering and exiting the MTO.
MODEL IMPLEMENTATION

- Use of jsprit java-based package which provides ample methods and constraint abilities to account for all scheduling considerations
FLEET-CENTERED APPROACH

- Optimized for Dual-Moves
- Staggered fleet dismissals
- Notifications via texts and push notifications:
  - Encapsulated scheduling process: truck drivers are notified ride-by-ride
  - Driver must notify MTO if their truck is delayed.
    - 1-hour turn around, algorithm ensures there is a reserve of unoccupied trucks at any given time in case of emergency
MODEL POTENTIAL

- Overlays of Google Maps onto solution space
- Broad Driver-MTO communication network with real-time locations and updates
- Integration of Microsoft bot APIs to update terminal conditions or handle tricky scheduling dilemmas
- Modularity allows for integration with rail and ocean transport
SUMMARY

- Robust core technology
- Scalable, user-friendly, and simplistic design allows for easy integration
- Core tech allows drivers and MTO operators to spend more time closing and less time waiting.