Dynamic Same-day Delivery with Crowd-shipping (in-store customers): Approximate Dynamic Programming Approach

Matthew Roorda
Kianoush Mousavi
Merve Bodur
Mucahit Cevik
Last-mile Delivery

• The very last step of the delivery process, from warehouse to customers

• It is the most expensive and time-consuming part of delivery process!

53%

Of Overall Shipping Cost
Trends in E-commerce

- E-commerce revenue is expected to increase 48% from 2018 to 2023 (Statista, 2019).

- The COVID-19 pandemic has accelerated e-commerce trends by around 5 years (IBM, 2020).

- 87% of logistics providers and retailer would make use of crowd-sourced delivery by 2028, compared to 30% in 2019 (Zebra, 2018).
Crowd-shipping

Sharing individuals’ spare time and/or vehicle capacity for delivering goods.
Example of Crowd-shipping Operation: (Archetti et. al., 2016)
Background and Motivation

➢ Humans are driven by Instant Gratification

➢ US same-day delivery market expects 21.6% annual growth from 2021-2026

➢ Proximity of brick-and-mortar stores to online customers

➢ Brick-and-mortar stores have access to a large pool of potential crowd-shippers (i.e., in-store customers)

➢ Incorporating uncertainty of crowd-shippers and online orders in operational decisions

➢ Scalable solution approaches for (near) real-time decision-making
Problem Description

- Employing in-store customers as crowd-shippers for delivering online orders
- The online orders should be delivered within few hours
- Crowd-shippers can deliver multiple orders but only can have two stops (their own destination and an intermediate stop)
- Cost penalty for unserved orders within the delivery deadline
- Crowd-shippers are compensated based on:
  - Number of delivered packages.
  - Their distance deviation
Problem Description

- Crowd-shippers and online orders arrival randomly through the operation horizon
- At each time period there is a new set of crowd-shippers
- Decisions at each epoch:
  - Assigning crowd-shippers to online orders
  - Whether to postpone the orders to the next time period
Problem Description

\[ T(t) = T(t-1) + \sum_{i=1}^{n} \Delta T_i \]

Time

\( t-1 \) \hspace{2cm} \( t \) \hspace{2cm} \( t+1 \)
Problem Description

\[ t - 1 \quad \rightarrow \quad t \quad \rightarrow \quad t + 1 \]

Time
Problem Description

The questions to be addressed at each decision epoch:

- To whom should we assign the online orders?
- Should we postpone the delivery in the hope for a future crowd-shipper with a lower compensation?
- Can we increase the total number of served orders by making smarter assignment decision?
- Should we wait for a crowd-shipper with a higher capacity to bundle the deliveries?
- How can we consider the down-stream uncertainty for real-time decision-making?

Dynamic Programming
Dynamic programming

Sequential decision-making problems can be difficult to solve via dynamic programming.

- Imagine the case below:

<table>
<thead>
<tr>
<th># Locations</th>
<th># orders</th>
<th>Delivery dead-time</th>
<th># crowd-shipper</th>
<th>Crowd-shipper capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>25</td>
<td>8</td>
<td>25</td>
<td>1 to 4</td>
</tr>
</tbody>
</table>

Total number of states: $9.5 \times 10^{58}$

Almost a billion times more than the number of atoms on Earth

Approximate Dynamic Programming
Overcoming three curses of dimensionality

• **Outcome Space:**
  - The concept of post-decision state is introduced.
  - Post-decision state is the state immediately after the action but before arrival of new random information.

• **State Space:**
  - The value function of post-decision state is approximated as linear function of online orders.

• **Action Space:**
  - Modelling assignment of online orders to crowd-shippers as a mixed-integer program.
ADP vs Myopic policy

- Value functions are calculated based on value iteration algorithm (Powell, 2011).
- Some Enhancements are incorporated in this algorithm:
  - Hierarchical Aggregation (George et al., 2008)
  - Monotonicity (Jiang & Powell, 2015)
- The final value functions are used for real-time decision making i.e., ADP policy.
- The Myopic Policy is to serve orders as soon as possible via solving the assignment problem.
Study area (Downtown Toronto, Canada)

Test instance characteristics:

• 117 locations.
• 52 time-periods (each 15 mins)
• Total orders= 1000
• Maximum Delivery deadline= 8 periods (i.e., 2 hours)
• Crowd-shippers capacity =1 to 4 packages (equal likelihood).
• Crowd-shippers deviation range ={1.1, 1.3, 1.5}
• Policy evaluations are based on simulation of 1000 sample paths.
Study area (Downtown Toronto, Canada)
Total Cost-saving (ADP vs. Myopic)

- Higher cost-saving with increase in number of crowd-shippers
- Higher cost-saving if we have crowd-shippers with higher deviation range

\[ \zeta = \text{Crowd-shippers deviation range} \]

\[ \frac{\text{Total number of crowd-shipper}}{\text{Total number of orders}} \]

\[ \text{Total cost saving (\%)} \]
Change (%) in number of served orders and assigned Crowd-shippers (ADP vs. Myopic)

- More prominent increase in number of served orders when there are fewer number of crowd-shippers!
- With Increase in number of crowd-shippers, the ADP policy focuses more on bundling to reduce the total cost!

\[ D = \text{Maximum delivery deadlines in time periods (8 time periods = 2 hours)} \]
Average time-period to serve the online orders (ADP vs. Myopic)

\[ D = \text{Maximum delivery deadlines in time periods (8 time periods = 2 hours)} \]

- The ADP policy requires one to two additional time-periods to serve the orders.
Results summary

ADP Policy VS. Myopic:
(based on extensive sensitivity analysis on the various model parameters (refer to the paper))

- Up to 25% decrease in total cost.
- Up to 10% increase in total delivered orders.
- Up to 21% increase in average number of assigned orders per crowd-shippers.
- Average additional postponement is about 1.5 time periods (for the case with delivery deadline = 8 time periods).
- Up to 65% decrease in average distance deviation of crowd-shippers.
- Percentage of total served orders ranges from 70% to 99%.
Conclusion

- A dynamic crowd-shipping model is introduced.
- An approximate dynamic programming algorithm is developed.
- A test instance based on downtown Toronto is generated.
- Promising results from ADP policy in comparison to myopic policy.
More about this work!

• More detailed model description and analysis can be found in: