MGSC662 Instacart

Chen, Shuxi
Wan, Boyang
Tang, Henry
Li, Jintao
Annigeri, Samarth

December 8, 2024

Introduction

Instacart is a fast-growing app that connects various grocery stores with customers, providing convenient grocery delivery services. Customers can shop on the mobile app and get their groceries delivered directly to their doors in as fast as 30 minutes, eliminating the need to visit multiple stores in person. Instacart operates within a complex four-sided marketplace that includes customers, shoppers, delivery drivers, and retail stores. Unlike other delivery platforms such as Uber Eats, Instacart's model presents unique optimization challenges due to its focus on grocery delivery, which involves higher business value and more complex requirements compared to other order delivery fields.

Instacart's business model is inherently complex, encompassing multiple interdependent processes that must be optimized to ensure operational efficiency. From assigning shoppers to orders, minimizing in-store shopping time, batching multiple orders, to determining the most efficient delivery routes, each process plays a critical role in the overall operational model. This project specifically focuses on **order dispatch optimization**, which involves assigning shoppers to orders based on proximity and timing, and **delivery routing optimization**, which determines the most efficient paths for shoppers to deliver orders from stores to customers. These two components are critical to enhancing shopper efficiency, reducing operational costs, and improving the overall customer experience.

This project focuses on the **Greater Montreal area** as the testing ground for our optimization model due to its unique urban characteristics. Montreal features a complex network of roads, including numerous one-way streets and construction zones, as well as neighborhoods with varying population densities and challenging seasonal weather conditions. These factors not only make delivery operations more intricate but also provide a realistic and rigorous environment to test and refine our model. By addressing delivery routing challenges specific to Montreal, this project aims to improve shopper efficiency and reduce delivery times. Furthermore, the insights gained from this dynamic urban setting can help create scalable solutions applicable to other cities with similar complexities.

Problem Description and Formulation

Since this project focuses on the processes of order dispatching and delivery routing optimization, we simplify and abstract the business scenario to concentrate on a defined peak period between **10:00 AM and 3:00 PM**, when the majority of orders are placed. These orders are characterized by specific attributes such as location coordinates and area types (residential, commercial, or suburban).

For the purpose of this project, we selected **Dollarama** as the supplier due to its well-distributed network across the Greater Montreal area. Dollarama's presence in diverse neighborhoods provides an ideal basis for modeling and testing delivery optimization. By abstracting the broader operational complexities of Instacart into this controlled scope, we aim to address the challenges of efficient shopper assignment and delivery routing.

Project Objective

The goal of this project is to develop an optimization model that **minimizes the delivery time per order during peak hours**, while adhering to operational constraints. The model achieves this by addressing two distinct stages of delivery operations: efficiently assigning shoppers to Dollarama locations during the pickup stage and determining the most effective delivery routes for transporting orders to customers. By focusing on these two stages, the project aims to enhance shopper efficiency, reduce delivery times, and create a scalable solution for Instacart's delivery operations. This problem is modeled as an **Open Vehicle Routing Problem (OVRP)**, reflecting real-world operations where shoppers transition seamlessly between tasks without returning to their starting point.

This project formulates an optimization model that separates the problem into two stages:

- **Pickup Stage Optimization:** Determines when and which shopper is sent to pick up orders from Dollarama locations.
- **Delivery Stage Optimization:** Determines the most efficient routes for delivering the picked-up orders to customers.

Pickup Stage Optimization

The **pickup stage** involves assigning shoppers to collect orders from Dollarama branches based on proximity and waiting time. When the model first runs, the process starts from an **initial point**, such as the shopper's home base or a designated starting location. After completing one delivery, the next pickup process starts from the **end point of the previous delivery**, ensuring continuity and efficiency in the optimization process.

The platform assigns shoppers to orders based on the following criteria:

- **Two Order Assignment:** If more than **two** orders are in the same area, the platform assigns the **nearest shopper** to pick them up together.
- **Waiting Time Priority:** For single orders waiting longer than **10 minutes**, a shopper is dispatched immediately to ensure timely fulfillment.

This stage focuses on efficiently allocating shoppers to pickup tasks while adhering to operational constraints. The dynamic reassignment of the starting point after each completed delivery ensures that the optimization model mimics **real-world conditions**, where shoppers seamlessly transition between tasks without unnecessary idle time or backtracking.

Delivery Stage Optimization

Once orders are picked up, the **delivery stage** focuses on determining the optimal routes to minimize delivery time. To achieve this, the model considers several critical factors:

- Each shopper delivers up to **two orders in a single trip** (batch limit)

- The city of Montreal is divided into **four distinct areas**—residential, commercial, mixed-use, and suburban—to account for geographic diversity and varying traffic conditions.
- Delivery routes incorporate **penalties for high-traffic zones, construction sites, and one-way roads** to reflect real-world constraints and challenges.
- For trips with multiple orders, the model optimizes the **sequence of deliveries**, determining which customer to serve first based on proximity and travel conditions.

Example Scenario

Scenario Overview

- A Dollarama branch located at **620 Rue Sainte-Catherine Ouest, Montreal** receives four orders during the peak period (10:00 AM to 3:00 PM). These orders have specific attributes, such as:
  - **Order A:** Residential area at 1200 Rue Bishop, 1.5 km from the store.
  - **Order B:** Commercial area at 1250 Boulevard René-Lévesque Ouest, 2.0 km from the store.
  - **Order C:** Suburban area at 5000 Avenue de Monkland, 5.0 km from the store.
  - **Order D:** Mixed-use area at 650 Boulevard De Maisonneuve Ouest, 1.2 km from the store.
- Two shoppers are available in the proximity, **Shopper X** and **Shopper Y**.
- The platform dynamically assigns orders based on proximity and waiting time.

Pickup Stage Execution

- **Step 1: Proximity-Based Assignment** Orders A (1200 Rue Bishop) and D (650 Boulevard De Maisonneuve Ouest) are located close to each other in the same area (1.5 km and 1.2 km from the store, respectively). Based on the proximity-based assignment rule, Shopper X, who is nearest to the Dollarama branch, is assigned to pick up both orders together.
- **Step 2: Waiting Time Priority** Order B (1250 Boulevard René-Lévesque Ouest) has been waiting for 12 minutes, exceeding the 10-minute threshold. Shopper Y, who is available nearby, is dispatched immediately to pick up Order B.
- **Step 3: Reassignment for the Remaining Order** After completing the first delivery, Shopper X becomes available and is reassigned to pick up Order C (5000 Avenue de Monkland), located in the suburban area (5.0 km from the store).

Delivery Stage Execution

- **Step 1: Sequence Optimization for Shopper X** Shopper X starts with Orders A (1200 Rue Bishop) and D (650 Boulevard De Maisonneuve Ouest). The delivery sequence is optimized based on proximity, with Order D delivered first, followed by Order A. This minimizes overall travel distance and time.
- **Step 2: Route Adjustment for Shopper Y** Shopper Y delivers Order B (1250 Boulevard René-Lévesque Ouest) in the commercial area. Since only one order is assigned, the delivery route is straightforward with no further adjustments.
- **Step 3: Delivery to Suburban Area for Shopper X** After completing Orders A and D, Shopper X proceeds to deliver Order C (5000 Avenue de Monkland) in the suburban area. This route incorporates penalties for one-way streets and traffic congestion to ensure realistic delivery time estimates.

Numerical Implementation and Results

Model Formulation

Notation	Category	Description
S	Set	Set of supplier locations (possible pickup points).
D	Set	Set of demand locations (possible customer locations).
I	Set	Set of driver initial locations.
Δ_S	Set	Set of pickup locations (only 1 element contained).
Δ_D	Set	Set of delivery locations (up to 2 elements contained).
T_{ij}	Parameter	Travel time between locations i and j, where $i, j \in S \cup D$.
x_{ij}	Decision Variable	Binary variable: 1 if driver k travels from i to j, 0 otherwise.

Table 1: Notation and Categories

For pickup routing, we define a set of binary decision variables x_{ij} that indicate whether the path includes the edge from node i to node j. The objective is to minimize the total travel time, computed as the sum of the edge travel times multiplied by these binary variables. Flow balance constraints ensure that exactly one unit of flow originates from the initial points and terminates at the supply center nodes in Δ_S. Additional constraints guarantee that only feasible routes are constructed and that we do not create disconnected sub-tours.

$$\min_{\{x_{ij}\}} \sum_j \sum_i T_{ij} \cdot x_{ij}$$

s.t.

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} \geqslant 0 \quad \forall i \in I$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} \leq 1 \quad \forall i \in I$$

$$\sum_{i \in I} (\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji}) = 1$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = 0 \quad \forall i \in N \backslash \{I \cup \Delta_s\}$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = -1 \quad i = m \in \Delta_S$$

For delivery routing, a similar shortest path formulation is used, but this time the flow starts at a chosen source (supply center) and ends at the given demand point(s). When a single demand point is involved, we solve a straightforward shortest path problem. When there are two demanders, the code tests both possible sequences and picks the route with the least total travel time.

$$\min_{\{x_{ij}\}} \sum_j \sum_i T_{ij} \cdot x_{ij}$$

s.t.

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = \begin{cases} 1 & i = m \\ -1 & i = n \\ 0 & \text{o. w.} \end{cases}$$

Data Simulation

To ensure the optimization model accurately reflects real-world delivery challenges in Montreal, we simulated input data incorporating both location-based variability and time-related factors that capture the city's unique characteristics.

- **Location-based variability** features 26 real Dollarama branches and 100 simulated demand points distributed across four land-use zones. Shoppers' initial points and customer locations were assigned to these demand sites, with each demand point paired to its nearest Dollarama branch.
- **Time-related factors** include traffic congestion, construction zones, and neighborhood-specific access restrictions, translated into additional time costs as penalties that modify the basic driving time for each edge in the network.

Location Setup

- **Dollarama Stores and Demand Locations:** To create a realistic representation of supplier locations, we incorporated data from 26 Dollarama branch stores across Montreal as the supplier set (S). The geographic coordinates of each store were utilized to define its location and associated delivery network. For demand, 100 demand destinations (D) were simulated across Montreal, taking into account the varying population densities and activity levels in different land-use zones. Recognizing that higher-density areas generate more demand, we assigned demand points proportionally based on four land-use patterns – residential, commercial, mixed-use, and suburbs – where busy or crowded areas have denser demand clusters, while suburban regions have sparser distribution. By anchoring demand points to land-use characteristics, the simulation captures the essential variability in customer density and delivery needs.

  The demand distribution and corresponding Dollarama store locations were categorized as follows:

Land Uses	**Percentage of Demand Points**	**Demand Sites (D)**	**Dollarama Stores (S)**
Residential	45%	D1–D45	S1–S9
Commercial	25%	D46–D70	S10–S17
Mixed-use	20%	D71–D90	S18–S23
Suburbs	10%	D91–D100	S24–S26

Table 2: Demand Distribution and Corresponding Dollarama Stores

- **Initial Points for Shoppers and Source Locations for Orders:** A total of **100 shoppers** and **299 orders** with timestamps logged between 10 a.m. and 3 p.m. are randomly assigned to demand location points within $D_1 \sim D_{100}$. For shoppers, the assigned demand locations serve as their **initial starting points**, reflecting their proximity to delivery tasks. For orders, the assigned demand locations represent the **customer locations**, ensuring that the spatial distribution of orders mirrors real-world delivery scenarios.
- **Demand-Supply Location Pairing and Batching Mechanism:** Within each land-use zone, we established designated demand-supply pairs to reflect the localized nature of delivery operations. Each Dollarama branch store (S) is associated with a specific subset of demand locations (D) based on geographic proximity. For example, in residential areas, a Dollarama branch S_3 may serve demand points clustered in its immediate vicinity, such as D_{42}, D_{18}, D_{26}. Each demand location is exclusively paired with a single Dollarama branch, ensuring a one-to-one mapping of demand to supply locations.

  This pairing forms the foundational unit for batching orders. When orders from demand locations associated with a specific branch S_i accumulate to two batches, the model assigns a shopper whose initial location is closest to S_i to pick up and deliver the batched orders.

Time Cost Setup

- **Basic Driving Time Between Locations (T_{ij}^{base}):** A driving time matrix was generated using the **Azure Bing Maps API**, providing estimated basic travel times between all nodes, including both store (S) and demand (D) locations.

- **Construction Zones Penalties:** To reflect the impact of construction on delivery operations, **1.5% of road edges** were randomly selected as affected by construction. Among these, **1% are passable construction routes**, introducing delays while remaining feasible, and **0.5% are blocked routes**, making them completely unavailable for travel. These proportions are small enough to maintain route feasibility but large enough to significantly impact the optimization.

  - **Road Closures**
    We set **0.5% of road edges** closed due to construction. The travel time (T_{ij}) is set to infinity ($T_{ij} = \infty$), ensuring they are excluded from feasible paths. If a route is incorrectly chosen to pass through a closed road, represented by the binary variable A_{ij}, an additional large penalty ($M = 10,000$) is applied to strongly discourage its selection. The adjusted travel time for such routes is calculated as:
    $$T_{ij} = T_{ij}^{\text{base}} + MA_{ij}$$

  - **Passable Construction Routes**
    We set **1% of road edges** as construction routes that are still passable but introduce significant delays. These routes are penalized to simulate the increased travel time due to construction. If a passable construction route is chosen, represented by the binary variable B_{ij}, the model adds a penalty ($M = 10,000$) to discourage unnecessary use. The travel time is adjusted as:
    $$T_{ij} = T_{ij}^{\text{base}} + MB_{ij}$$

- **One-Way Streets Penalties:**

  To account for the restrictions imposed by one-way streets, **0.5% of road edges** were randomly selected to represent one-way traffic rules. For routes that violate these rules, the travel time (T_{ij}) is set to infinity ($T_{ij} = \infty$), making these paths infeasible.

  If a route passing through a one-way street in the wrong direction is selected, represented by the binary variable C_{ij}, the model applies a significant penalty ($M = 10,000$) to strongly discourage its use. The adjusted travel time for such routes is calculated as:
  $$T_{ij} = T_{ij}^{\text{base}} + MC_{ij}$$

- **Distance Weighting:**

  Travel times in urban areas are generally slower compared to suburban regions, largely due to higher traffic volumes and more intricate road networks. Urban areas often experience increased friction, driven by greater activity levels such as more vehicles, pedestrians, and complex infrastructure like traffic signals and intersections. To accurately capture this variation, we introduced a **distance weighting factor** that applies proportional driving time costs based on the combination of land-use zones for each edge in the network.

  The weighting schema for different land-use combinations is outlined below:

Land-Use Combination	**Residential**	**Commercial**	**Mixed-use**	**Suburbs**
Residential	Rand (0.05, 0.15)	0.15	0.1	0.05
Commercial	0.15	Rand (0.15, 0.25)	0.15	0.1
Mixed-use	0.1	0.15	Rand (0.1, 0.2)	0.05
Suburbs	0.05	0.1	0.05	0

Table 3: Weighting Schema for Land-Use Combinations

The travel time for each edge is adjusted using the formula:

$$T_{ij} = T_{ij}^{\text{base}} \times (1 + \beta_{ij} \text{ urban friction})$$

- **Overall Travel Time Calculation:**

  The overall travel time T_{ij} for each edge in the network incorporates multiple factors to reflect real-world delivery challenges in Montreal. It is computed as:

$$T_{ij} = T_{ij}^{\text{base}} \times (1 + \beta_{ij} \text{ urban friction}) + MA_{ij} + MB_{ij} + MC_{ij}$$

Results and Interpretation

The final table presents each batch (See Table 4), along with:

- **Timestamp and Supply Center:** The departure time and origin of the batch.
- **Demanders:** The unique demand points served.
- **pickup_time, delivery_time, total_time:** Travel durations for each phase.
- **Shortest Pickup Path and Shortest Delivery Path:** The computed routes minimizing travel time.

These results show a clear pattern: when orders are grouped together at the same supply center within a short time frame, deliveries tend to be more efficient. Serving multiple demand points at once, rather than one at a time, creates a more consolidated route that usually cuts down on total travel time per order. On the flip side, if the orders are spread out over longer distances or arrive too far apart, it naturally takes more time to get them all delivered.

This makes sense intuitively. Clusters of nearby demand points are simpler to handle and require less travel, while scattered or numerous locations are more challenging and lead to longer trips. The batching rule strikes a balance between sending out deliveries immediately and waiting to combine them, reflecting a real-world trade-off in logistics planning.

All in all, the outcomes look reasonable and line up with what we'd expect from the distances, timing, and rules we've put in place. The routes and times calculated fit well with how such distribution scenarios play out in practice.

Time	S	D	P_time	D_time	Total Time	Pickup Path	Delivery Path
11:11:35	S3	['D28']	5.3	16	21.3	[('D20', 'S3')]	['S3', 'D28']
11:13:17	S1	['D11']	3.6	19.5	23.1	[('D6', 'S1')]	['S1', 'D11']
11:18:21	S3	['D30', 'D28']	5.3	25.5	30.8	[('D20', 'S3')]	['S3', 'D30', 'D28']
11:22:51	S1	['D13', 'D1']	3.6	27.9	31.5	[('D6', 'S1')]	['S1', 'D1', 'D13']
11:25:20	S2	['D26', 'D19']	2.5	26.8	29.3	[('D19', 'S2')]	['S2', 'D19', 'D26']
11:37:21	S3	['D33']	5.3	10.4	15.7	[('D20', 'S3')]	['S3', 'D33']
11:42:02	S1	['D11', 'D9']	3.6	41.6	45.2	[('D6', 'S1')]	['S1', 'D11', 'D9']
11:42:31	S2	['D20']	2.5	7.2	9.7	[('D19', 'S2')]	['S2', 'D20']
11:44:50	S2	['D18', 'D18']	2.5	7.1	9.6	[('D19', 'S2')]	['S2', 'D23', 'D18']
11:51:06	S1	['D10', 'D15']	3.6	45.4	49	[('D6', 'S1')]	['S1', 'D15', 'D10']
11:57:26	S1	['D16', 'D4']	3.6	35.5	39.1	[('D6', 'S1')]	['S1', 'D16', 'D4']
12:07:38	S3	['D29']	5.3	19.4	24.7	[('D20', 'S3')]	['S3', 'D29']
12:12:46	S1	['D14', 'D5']	3.6	42.5	46.1	[('D6', 'S1')]	['S1', 'D14', 'D5']
12:13:06	S2	['D21', 'D22']	2.5	13.5	16	[('D19', 'S2')]	['S2', 'D22', 'D17', 'D21']
12:15:45	S4	['D34']	2.4	23.1	25.5	[('D4', 'S4')]	['S4', 'D34']

Table 4: Pickup and Delivery Times with Shortest Paths

Problem Extensions

Time Window Integration

The current optimization model focuses primarily on minimizing travel time and efficiently allocating resources. However, it does not account for a critical aspect of real-world delivery operations: customer-specified delivery time preferences, or **time windows**. In Instacart's real-world service model, customers often specify preferred delivery intervals to align with their schedules. Incorporating these time windows into the optimization model can make the system more customer-centric and competitive, improving satisfaction and loyalty.

Proposed Adjustment We propose introducing a conceptual extension to the model by prioritizing deliveries based on **time window constraints**. This involves defining a **time window metric** that measures the gap between the time an order is placed and the customer's desired delivery time. Delivery routes would then be adjusted dynamically to ensure that orders nearing the end of their time windows are prioritized while still considering overall route efficiency. By integrating this logic, the model would balance operational efficiency with meeting customer-specific preferences.

Expected Outcomes Adding time window constraints would likely result in a small trade-off between efficiency and customer satisfaction. While the average delivery time might increase slightly due to prioritization, customers would benefit from receiving their orders within their specified intervals. This trade-off aligns with Instacart's strategic objectives of retaining existing customers and attracting new ones by offering a more personalized service.

Benefits of Time Window Integration Incorporating time windows into the model provides several advantages:

- **Customer Satisfaction:** Meeting delivery intervals enhances trust and strengthens customer relationships.
- **Operational Realism:** Reflects real-world practices, making the model more applicable to Instacart's business environment.
- **Competitive Edge:** Differentiates Instacart from competitors by aligning deliveries more closely with customer needs.
- **Long-Term Value:** Promotes repeat business through improved customer experiences.

Challenges and Considerations While this extension conceptually enhances the model, its practical implementation in urban settings such as Montreal introduces challenges. Factors like frequent construction projects, one-way streets, and high-density traffic zones often cause delays, complicating adherence to strict time constraints. A robust system would need to dynamically adjust for these variables, incorporating penalties for delays and prioritizing routes that minimize disruption to delivery schedules. Although these challenges exist, addressing them would make the model more realistic and adaptable.

Multi-Supplier and Multi-Store Demand Sharing

The current model assumes a one-to-one relationship between demand points and their nearest Dollarama store, focusing on a single supplier. While this simplification is useful for initial optimization, it overlooks the complexities of real-world delivery scenarios. Customers' orders often span multiple suppliers, such as Provigo, Walmart, and local markets, creating a many-to-many relationship between demand points, suppliers, and delivery drivers.

Proposed Adjustment To better reflect these realities, we propose extending the model to incorporate **multi-supplier and multi-store demand sharing**. In this extension, drivers are permitted to pick up orders from multiple suppliers during a single trip and are not required to complete all pickups before starting deliveries. A new constraint is introduced to ensure that deliveries to a demand point occur only after all required items for that point have been picked up from their respective suppliers. Specifically, the delivery time at a demand point must be greater than the latest pickup time from any supplier for that order, plus the travel time required between the supplier and the demand point.

Expected Benefits Incorporating multi-supplier demand sharing provides several operational and strategic advantages:

- **Increased Flexibility:** Allows drivers to dynamically adjust their routes to prioritize high-priority deliveries while completing pickups from other suppliers en route.
- **Improved Efficiency:** Reduces overall delivery time and optimizes the use of drivers and vehicles.
- **Cost Reduction:** Minimizes operational costs by consolidating pickups and deliveries more effectively.
- **Customer-Centric Model:** Enhances customer satisfaction by ensuring timely deliveries, even for multi-supplier orders.

Challenges and Practical Considerations Implementing this extension in a dynamic urban setting such as Montreal introduces several challenges. Drivers must navigate diverse routes that account for multiple supplier locations, traffic conditions, and urban constraints such as construction and one-way streets, significantly increasing routing complexity. Additionally, ensuring that all required items for a demand point are picked up in time to meet delivery deadlines adds another layer of complexity, as it requires precise time synchronization between pickups and deliveries. Furthermore, aligning operations across multiple suppliers necessitates robust communication and platform integration, making supplier coordination a critical factor for the success of this extension.

Contextual Relevance to Montreal Montreal's unique mix of grocery and retail suppliers, ranging from large chains to locally-owned markets, highlights the importance of a flexible service model. The city's bilingual population and diverse communities further emphasize the need to accommodate varying customer preferences. By incorporating multi-supplier demand sharing, the model adapts to these complexities, offering a scalable solution for diverse urban environments.

Recommendations and Conclusions

Recommendations If acting as a consultant for this project, the primary recommendation would be to leverage advanced optimization tools or machine learning techniques could significantly enhance the model's ability to handle larger problem instances. For example, heuristic methods or reinforcement learning algorithms could be employed to find near-optimal solutions efficiently for large-scale scenarios. These tools would also enable dynamic adjustments to the model, such as adapting to changing traffic conditions or incorporating **real-time order** data.

Lessons Learned During the modeling phase, we adopted innovative approaches, such as using the analogy of a water tube to conceptualize and refine our constraints. This helped us frame the relationships between demand points, suppliers, and drivers in a structured way, making it easier to translate abstract ideas into mathematical formulations. Additionally, significant effort was dedicated to reviewing academic literature and existing research in the field. By building on prior works, we not only validated our approach but also identified areas where our model could push the boundaries of current optimization practices.

This project underscored the significance of carefully designing the data simulation process and constructing a robust set of operational rules to reflect real-world delivery scenarios. By simulating diverse demand points, supplier locations, and urban constraints such as construction zones and traffic, we were able to create a realistic testing environment.

Future Improvements If revisiting this project, a key focus would be on identifying and developing unique contributions to differentiate our model from existing solutions. This is particularly important as we believe the optimization techniques implemented in this project, such as routing and delivery time minimization, are already being utilized by platforms like Instacart. Our approach primarily aimed to replicate a simplified version of their operations. In hindsight, conducting more thorough preliminary research to uncover specific methodologies and algorithms already in use by Instacart would have allowed us to focus on areas where we could innovate and add value.

Bottlenecks and Limitations A key limitation of this project was the difficulty in applying the model to real-life data. While the optimization framework worked well with simulated data, real-world datasets would introduce additional complexity, such as incomplete or inconsistent information, making implementation more challenging. Bridging this gap would require access to high-quality, real-time data and additional preprocessing efforts to ensure the model's applicability in practical settings.

Conclusion: Impact on Stakeholders

- **Customers:** The optimized delivery model ensures faster and more reliable grocery delivery, providing a seamless and convenient shopping experience.
- **Drivers:** Drivers benefit from optimized routes, reducing frustration due to construction or traffic while increasing earning potential through efficient order batching.
- **Instacart:** The platform achieves greater operational efficiency, enhancing customer loyalty and driving growth by differentiating itself from competitors.
- **Environment:** Optimized delivery routes reduce unnecessary travel, cutting down on fuel consumption and emissions, contributing to a more sustainable urban delivery model.

References

1. Gurobi Optimization. (n.d.). *Instacart: Delivering optimal shopping experiences.* Retrieved from `https://www.gurobi.com/case_studies/instacart-delivering-optimal-shopping-experiences/`

2. Gurobi. (2022, September 1). *How Instacart uses Gurobi optimization to maximize efficiency* [Video]. YouTube. Retrieved from `https://youtu.be/-Y8GGgT4Qf8?si=proCbUmOM7Ge5JYi`

3. Gurobi. (2022, September 20). *Optimization insights for delivery services* [Video]. YouTube. Retrieved from `https://youtu.be/F0iaBmYyLCI?si=ObmS5unN-2vhCs1U`

4. Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2007). *A survey on pickup and delivery problems, Part I: Transportation between customers and depot. Journal für Betriebswirtschaft.*