Final Project

BAN 630 – Optimization for Analytics

<u>Group 3</u>

Khush Domadiya

Dharmi Kanth Cigiri

Mohanasundaram Murugesan

Amisha Farhana Shaik

Gurobi Burrito Optimization Game

The objective of this project is to determine the optimal deployment of food trucks to maximize profit while serving customer demand for burritos. Using Gurobi, we formulate an optimization model considering truck setup costs, ingredient costs, and demand fulfillment constraints. The model ensures that every demand location is served by exactly one truck while minimizing costs and maximizing revenue.

Problem Definition

Decision Variables:

- *Zi*: Binary variable, 1 if truck is open, 0 otherwise.
- *Xij*: Binary variable, 1 if truck serves demand node, 0 otherwise.

Objective Function:

- Maximize profit = (Revenue from burrito sales) - (Cost of opening trucks).

Constraints:

- Each demand node must be served exactly once.
- A truck must be open if it serves any demand node.
- Profit is affected by ingredient costs and truck setup costs.

Dataset Overview

Day 1 round 2

File Name: round1-day2_demand_node_data.csv
What It Has - Information about demand nodes (customer locations) where burrito demand exists.

Columns and Definitions:

- Index - Unique identifier for each demand node. Example: demand2, demand11, etc.
- Name - Descriptive name of the demand node (e.g., a business or location name).
- X - X-coordinate of the demand node's location. Example: 190.7207
- Y - Y-coordinate of the demand node's location. Example: 72.6611
- Demand - Total number of burritos required at the demand node. Example: 20, 65, etc.

File Name: round1-day2_truck_node_data (1).csv
What It Has - Information about potential truck locations where food trucks can be deployed.

Columns and Definitions:

- Index - Unique identifier for each truck node. Example: truck2, truck8, etc.
- X - X-coordinate of the truck node's location.
- Y - Y-coordinate of the truck node's location.

File Name: round1-day2_demand_truck_data.csv
What It Has - The relationship and feasibility between each truck node and each demand node, indicating how many burritos a truck can serve to a given demand node and the distance between them.

Columns and Definitions:

- demand_node_index - Identifier for the demand node (should match the "index" in the demand node file).
- truck_node_index - Identifier for the truck node (should match the "index" in the truck node file).
- Distance - The distance between the truck node and the demand node.
- scaled_demand - The effective number of burritos that the truck can serve to the demand node. This value adjusts or limits the full demand based on feasibility (e.g., due to distance or capacity constraints).

File Name: round1-day2_problem_data.csv
What It Has:

Global problem parameters used in the optimization model, such as pricing and costs.

Columns and Definitions:

- burrito_price - The selling price of one burrito.
- ingredient_cost - The cost of ingredients for one burrito.
- truck_cost - The fixed cost incurred to open or deploy a truck at a truck node.

Optimization Model Implementation

The optimization model was implemented in Python using Gurobi. The following steps were followed:

Data Preprocessing:

- Loaded CSV files into Pandas DataFrames.
- Created dictionaries for demand, truck locations, and feasibility constraints.

Building Scaled Demand and Valid Pairs:

- Iterated through demand-truck data to check if scaled_demand i,j is positive.
- If valid, added the pair (i,j) to the scaled_demand dictionary and recorded it in the valid_pairs list.
- This filtering ensures that only feasible truck-demand assignments are considered, improving model efficiency.

Defining Decision Variables:

- Created binary decision variables xi,j for each valid pair (i,j), ensuring only feasible truck-demand assignments are modeled.

- xi,j = 1 means truck is assigned to demand node j, serving the exact number of burritos as per scaled_demand i,j.

In Summary - This process is not redundant; it's a two-step modeling approach:

- First, identify feasible assignments based on your data.
- Then, build the optimization model only using those feasible assignments.

<u>Summary of the Constraint with a Concrete Example</u>

Imagine we run a burrito truck business. We have three customer locations: demand1, demand2, and demand3. And suppose:

Valid truck assignments (from our data):

- demand1 can be served by truckA and truckB.
- demand2 can be served by truckA only.
- demand3 can be served by truckC and truckD.

For demand1, the code does:

- Create a list: trucks_that_can_serve_demand1 = [(truckA, demand1), (truckB, demand1)].
- It then adds the constraint - x[(truckA,demand1)]+x[(truckB,demand1)]=1
  Interpretation: Exactly one truck out of truckA and truckB must be assigned to serve demand1.

For demand2, the code does:

- List: trucks_that_can_serve_demand2 = [(truckA, demand2)].
- Constraint - x[(truckA,demand2)]=1x[(truckA, demand2)] = 1.
  Interpretation: TruckA must serve demand2 (since it's the only option).

For demand3, suppose:

- List: trucks_that_can_serve_demand3 = [(truckC, demand3), (truckD, demand3)].
- Constraint - x[(truckC,demand3)]+x[(truckD,demand3)]=1
  Interpretation: Exactly one truck between truckC and truckD is chosen to serve demand3.

<u>Why This is Critical</u>

Ensures Full Coverage: Every demand node gets exactly one truck serving it. This prevents any customer location from being served by multiple trucks (which might be redundant or inefficient) or by no trucks (which would leave customers unsatisfied).

Simplifies Decision Making: The solver doesn't have to choose between conflicting options—each customer location gets one clear assignment.

Scenario Analysis

Case 1: Base case.

- burrito truck cost: $250
- ingredient cost : $5
- price of burrito : $10 , Thus profit from each burrito is $5
- optimal profit is $3710 from 6 trucks

Truck	Demand Nodes Served	Total Scaled Burritos
Truck8	demand0, demand1, demand2, demand15, demand16, demand52, demand53	149
Truck16	demand8, demand11, demand12, demand13, demand14, demand18, demand20, demand21, demand51	214
Truck26	demand17, demand19, demand28, demand29, demand30, demand31, demand45	188
Truck37	demand27, demand33, demand41	170
Truck45	demand37, demand38, demand40	117
Truck56	demand22, demand23, demand25, demand26, demand35	204

```
Solution count 2: 3710 -385

Optimal solution found (tolerance 1.00e-04)
Best objective 3.710000000000e+03, best bound 3.710000000000e+03, gap 0.0000%

Optimal Profit = 3710.00

Chosen Truck Locations: ['truck8', 'truck16', 'truck26', 'truck37', 'truck45', 'truck56']
  Truck truck8 serves: ['demand0', 'demand1', 'demand2', 'demand15', 'demand16', 'demand52', 'demand53'], tot
al scaled burritos = 149
  Truck truck16 serves: ['demand8', 'demand11', 'demand12', 'demand13', 'demand14', 'demand18', 'demand20',
'demand21', 'demand51'], total scaled burritos = 214
  Truck truck26 serves: ['demand17', 'demand19', 'demand28', 'demand29', 'demand30', 'demand31', 'demand45'],
total scaled burritos = 188
  Truck truck37 serves: ['demand27', 'demand32', 'demand33', 'demand41'], total scaled burritos = 170
  Truck truck45 serves: ['demand37', 'demand38', 'demand40'], total scaled burritos = 117
  Truck truck56 serves: ['demand22', 'demand23', 'demand25', 'demand26', 'demand35'], total scaled burritos =
204
```

We have attained the same optimal solution in base case.

Case 2: Supply chain issue - Burrito ingredients price shoots up to $7

- burrito truck cost : $250
- ingredient cost : $7
- price of burrito : $10 , Thus profit from each burrito is $3

Truck	**Demand Nodes Served**	**Total Scaled Burritos**
Truck14	demand0, demand1, demand2, demand12, demand14, demand15, demand16, demand17, demand18, demand19, demand20, demand52	215
Truck18	demand8, demand11, demand13, demand21, demand22, demand23, demand51, demand53	247
Truck26	demand28, demand29, demand30, demand31, demand45	162
Truck33	demand25, demand26, demand35, demand37, demand38, demand40	194
Truck37	demand27, demand32, demand33, demand41	170

```
Optimal Profit = 1714.00

Chosen Truck Locations: ['truck14', 'truck18', 'truck26', 'truck33', 'truck37']
  Truck truck14 serves: ['demand0', 'demand1', 'demand2', 'demand12', 'demand14', 'demand15', 'demand16', 'de
mand17', 'demand18', 'demand19', 'demand20', 'demand52'], total scaled burritos = 215
  Truck truck18 serves: ['demand8', 'demand11', 'demand13', 'demand21', 'demand22', 'demand23', 'demand51',
'demand53'], total scaled burritos = 247
  Truck truck26 serves: ['demand28', 'demand29', 'demand30', 'demand31', 'demand45'], total scaled burritos =
162
  Truck truck33 serves: ['demand25', 'demand26', 'demand35', 'demand37', 'demand38', 'demand40'], total scale
d burritos = 194
  Truck truck37 serves: ['demand27', 'demand32', 'demand33', 'demand41'], total scaled burritos = 170
```

Case 3: We the fast-food chain and reduce ingredients cost by a $1. Then optimal profit spikes.

- burrito truck cost : $250
- ingredient cost : $4
- price of burrito: $10 , Thus profit from each burrito is $6

Truck	**Demand Nodes Served**	**Total Scaled Burritos**
Truck8	demand0, demand1, demand2, demand15, demand16, demand52, demand53	149
Truck16	demand8, demand11, demand12, demand13, demand14, demand18, demand20, demand21, demand51	214
Truck26	demand17, demand28, demand29, demand30, demand31, demand45	188
Truck37	demand27, demand32, demand33, demand41	170
Truck45	demand37, demand38, demand40	117
Truck56	demand22, demand23, demand25, demand26, demand35	204

```
Optimal Profit = 4752.00

Chosen Truck Locations: ['truck8', 'truck16', 'truck26', 'truck37', 'truck45', 'truck56']
  Truck truck8 serves: ['demand0', 'demand1', 'demand2', 'demand15', 'demand16', 'demand52', 'demand53'], tot
al scaled burritos = 149
  Truck truck16 serves: ['demand8', 'demand11', 'demand12', 'demand13', 'demand14', 'demand18', 'demand20',
'demand21', 'demand51'], total scaled burritos = 214
  Truck truck26 serves: ['demand17', 'demand19', 'demand28', 'demand29', 'demand30', 'demand31', 'demand45'],
total scaled burritos = 188
  Truck truck37 serves: ['demand27', 'demand32', 'demand33', 'demand41'], total scaled burritos = 170
  Truck truck45 serves: ['demand37', 'demand38', 'demand40'], total scaled burritos = 117
  Truck truck56 serves: ['demand22', 'demand23', 'demand25', 'demand26', 'demand35'], total scaled burritos =
204
```

Case 4: worse case supply chain crisis - Near break-even strategy

- burrito truck cost : $250
- ingredient cost : $9
- price of burrito : $10 , Thus profit from each burrito is $1
- optimal profit is $27

Truck	**Demand Nodes Served**	**Total Scaled Burritos**
Truck15	demand0, demand1, demand2, demand8, demand11, demand12, demand13, demand14, demand15, demand16, demand18, demand19, demand20, demand21, demand51, demand52, demand53	294
Truck26	demand17, demand27, demand28, demand29, demand30, demand31, demand32, demand33, demand41, demand45	250
Truck33	demand22, demand23, demand25, demand26, demand35, demand37, demand38, demand40	233

```
Optimal Profit = 27.00

Chosen Truck Locations: ['truck15', 'truck26', 'truck33']
  Truck truck15 serves: ['demand0', 'demand1', 'demand2', 'demand8', 'demand11', 'demand12', 'demand13', 'dem
and14', 'demand15', 'demand16', 'demand18', 'demand19', 'demand20', 'demand21', 'demand51', 'demand52', 'dema
nd53'], total scaled burritos = 294
  Truck truck26 serves: ['demand17', 'demand27', 'demand28', 'demand29', 'demand30', 'demand31', 'demand32',
'demand33', 'demand41', 'demand45'], total scaled burritos = 250
  Truck truck33 serves: ['demand22', 'demand23', 'demand25', 'demand26', 'demand35', 'demand37', 'demand38',
'demand40'], total scaled burritos = 233
```

Case 5: Minimum possible loss - Getting away with least loss when

- burrito truck cost : $250
- ingredient cost : $9.1
- price of burrito : $10 , Thus profit from each burrito is $0.99
- optimal profit is $-50 or $50 loss

Truck	**Demand Nodes Served**	**Total Scaled Burritos**
Truck15	demand0, demand1, demand2, demand8, demand11, demand12, demand13, demand14, demand15, demand16, demand18, demand20, demand21, demand51, demand52, demand53	279
Truck26	demand17, demand19, demand27, demand28, demand29, demand30, demand31, demand32, demand33, demand41, demand45	265
Truck33	demand22, demand23, demand25, demand26, demand35, demand37, demand38, demand40	233

```
Best objective -5.070000000000e+01, best bound -5.070000000000e+01, gap 0.0000%

Optimal Profit = -50.70

Chosen Truck Locations: ['truck15', 'truck26', 'truck33']
  Truck truck15 serves: ['demand0', 'demand1', 'demand2', 'demand8', 'demand11', 'demand12', 'demand13', 'dem
and14', 'demand15', 'demand16', 'demand18', 'demand20', 'demand21', 'demand51', 'demand52', 'demand53'], tota
l scaled burritos = 279
  Truck truck26 serves: ['demand17', 'demand19', 'demand27', 'demand28', 'demand29', 'demand30', 'demand31',
'demand32', 'demand33', 'demand41', 'demand45'], total scaled burritos = 265
  Truck truck33 serves: ['demand22', 'demand23', 'demand25', 'demand26', 'demand35', 'demand37', 'demand38',
'demand40'], total scaled burritos = 233
```

Key Observations & Insights

Break-even Truck Utilization:

When the ingredient cost increases, the profit per burrito decreases, requiring more sales to cover truck setup costs.

- At $5 profit per burrito, a truck needs to sell 50 burritos to break even.
- At $3 profit per burrito, a truck needs to sell 84 burritos to break even.
- At $1 profit per burrito, the model minimizes the number of trucks to 3, reducing costs.

Impact of Ingredient Cost Reduction:

- Lowering ingredient costs increases profit margins but does not significantly change the number of trucks used.
- If the ingredient cost drops by $1, optimal profit spikes while maintaining the same truck assignments.

Worse Case Scenario - Near Break-even:

- With a $9 ingredient cost, the company can still make a small profit ($27) by reducing the number of trucks to 3.
- If the cost increases to $9.1, the company enters a loss-making zone.

Recommendations & Future Considerations

- Cheap or smaller trucks: trucks costing $150 or below to setup (with limited scaling capacity i.e. ability to serve demand) can be setup get more profit. Analysis by simulation needed before setting up
- Value meals: priced at $7 with a profit of $4 can be introduced which could gain more popularity. Analysis by simulation needed before setting up
- Kids meals: This code make the problem nonlinear as kid meals might boost sales of normal burritos and gain more profit