**Optimizing Supply Chain Logistics with Genetic Algorithms**

**Table of Contents**

[Optimizing Supply Chain Logistics with Genetic Algorithms 3](#_Toc153178516)

[Executive Summary: 3](#_Toc153178517)

[1. Introduction: 3](#_Toc153178518)

[2. Dataset Overview: 3](#_Toc153178519)

[3. Optimization Model: 4](#_Toc153178520)

[4. Anticipated Results: 8](#_Toc153178521)

[5. Analysis of Results: 9](#_Toc153178522)

[6. Conclusion: 10](#_Toc153178523)

# Optimizing Supply Chain Logistics with Genetic Algorithms

# Executive Summary:

This project aims to optimize supply chain logistics by utilizing Genetic Algorithms (GAs) in conjunction with python gurobi optimizer learned in the course in order to reduce costs, maximize operational efficiency, and guarantee on-time customer deliveries. This project's dataset, which closely mimics actual supply chain scenarios, includes important variables including product demand, initial inventory, transportation costs, and delivery time limits.

# Introduction:

In the light of today’s dynamic business environment, organizations seeking to achieve competitive advantage in the market, optimizing their supply chain processes is crucial in achieving this goal. Supply chain optimization for efficient logistics play a very important role in meeting customer demands promptly and the business is able to increase revenue by reducing cost of running their supply chain logistics. This project explores the application of Genetic Algorithms, inspired by natural selection processes, to come up with some of the best solutions to optimize supply chain logistics. After the genetic algorithms find a viable solution Python Gurobi optimization takes over and further optimizes the the selected solution until a feasible solution is found. The optimization used herein sufficiently tackles the complexities associated with supply chain optimization. The company used in this project was given a name, Masters Logistics.

# 2. Dataset Overview:

The optimization model is based primarily on the dataset used in this research.

**Product demand**: which represents the market's need for particular products, is one of its components.

**Initial Inventory:** The quantity of each product's stock at launch.

**Transportation costs:** The prices incurred when moving goods from one place to another.

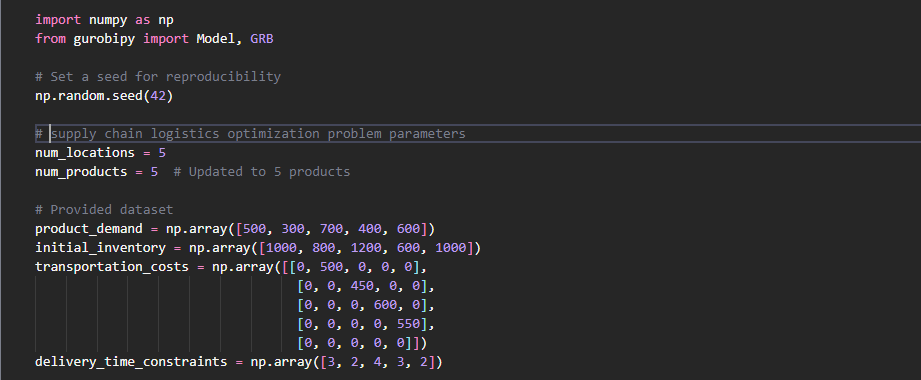
**Delivery Time Constraints:** Limitations on the amount of time needed to transport goods between locations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product | Demand (units) | Initial Inventory (units) | Transportation Cost ($) | Delivery Time Constraint (days) |
| A | 500 | 1000 | 500 | 3 |
| B | 300 | 800 | 450 | 2 |
| C | 700 | 1200 | 600 | 4 |
| D | 400 | 600 | 550 | 3 |
| E | 600 | 1000 | 420 | 2 |

# 3. Optimization Model:

This project's main focus is on applying genetic algorithms to solve challenging, non-linear supply chain optimization challenges. Genetic algorithms evolve possible solutions via generations in a manner similar to natural selection. The python Gurobi optimization then further optimizes the genetic algorithm’s solutions providing the best solution that meets the objective of the model. This pertains to supply chain logistics and entails figuring out how best to distribute products across different regions. The company used in this project has five locations and specializes in five products.

Below is a screen shot of the data representation in the model.



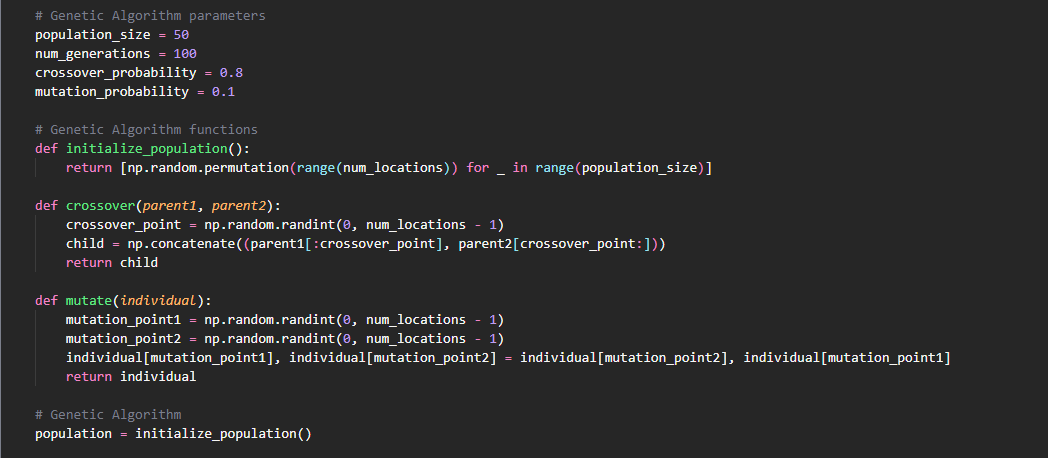
**The optimization model's decision variables consist of the following:**

The number of locations is a representation of the various supply chain network nodes.

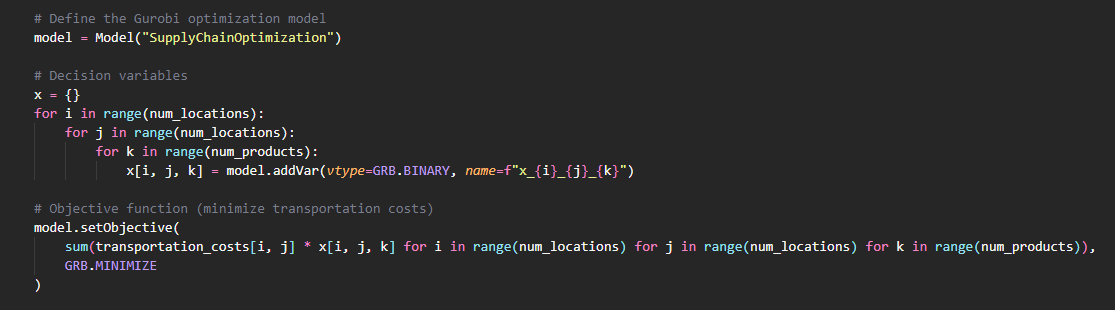
Number of Products: Indicates how many different items are being shipped.

The goal of the target function is to reduce transportation expenses while taking into account the price of transferring each commodity between various locations. Product demand, inventory restrictions, and delivery time limits are all taken into account by the model.

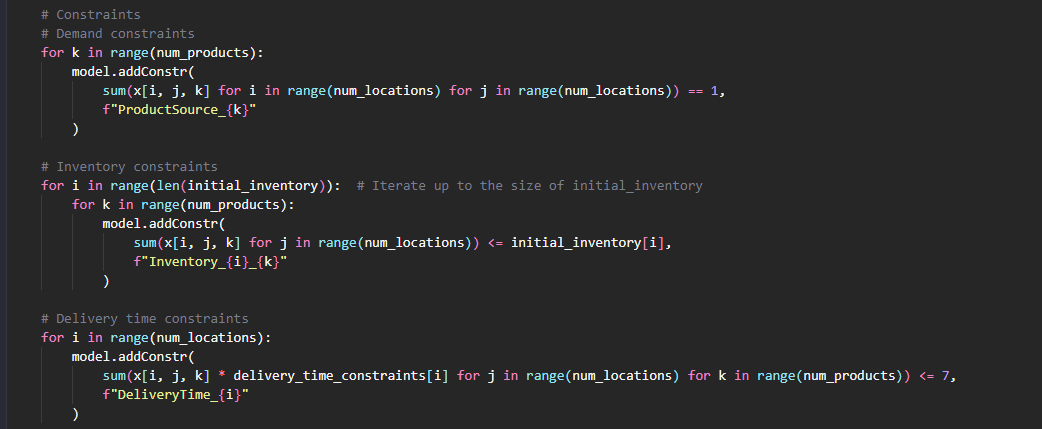
Attached below is a screenshot of the genetic algorithm initialization and functions.

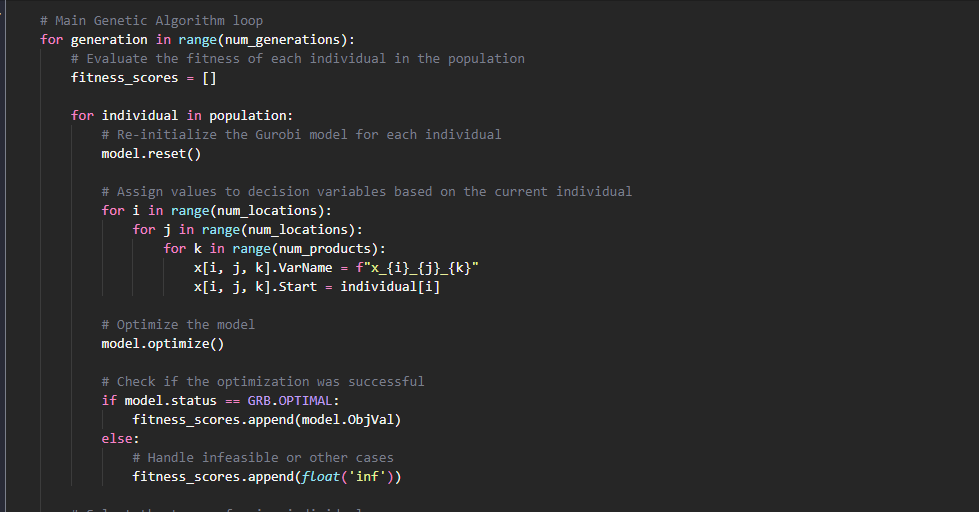


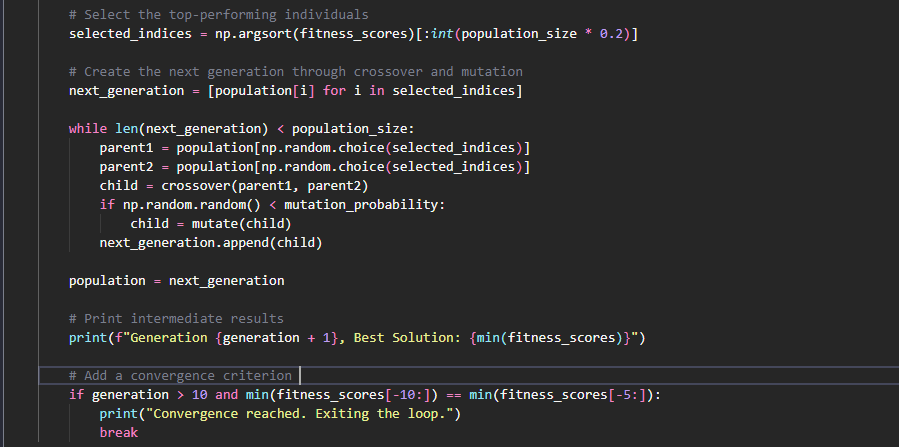
Attached below is the gurobi optimization initialization code snippet screenshot.

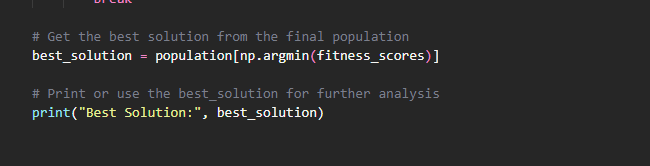


Attached below is the Gurobi model constraints code snippet screenshot.



Attached are screenshots of the main loop of the model.  






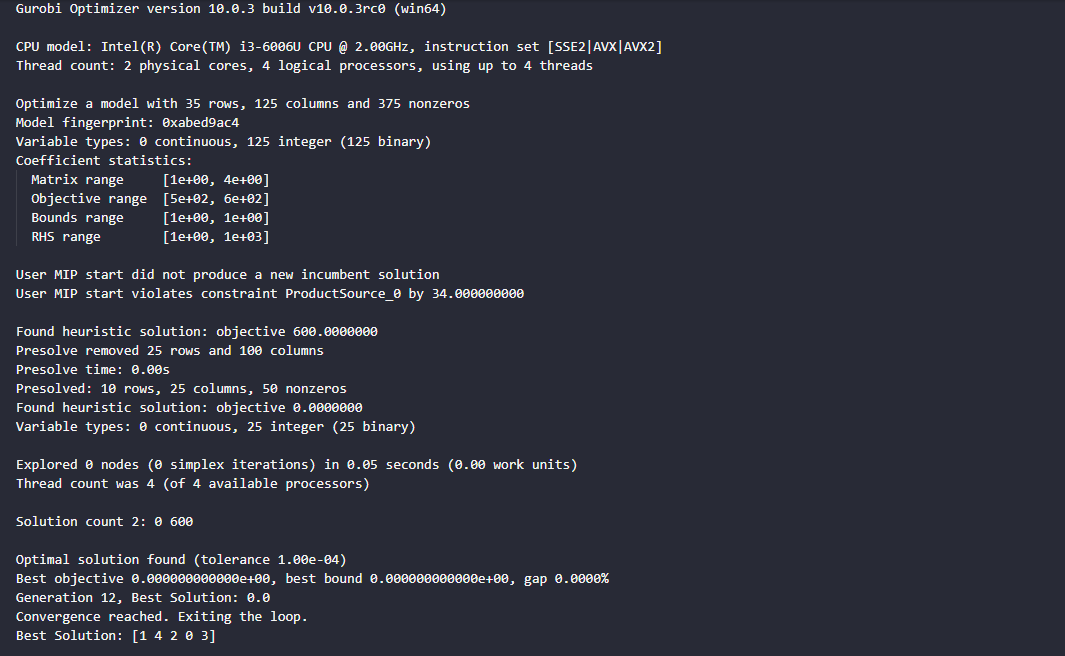
# 4. Anticipated Results:

The expected results include the best and optimal location to source each product. The projected outcomes include multiple critical performance indicators:

Transportation Cost Management: By effectively allocating items to destinations, the approach is anticipated to reduce transportation costs.

Better Delivery Schedules: In order to satisfy customers, the optimal solution should help to improve delivery schedules.

Resource Utilization: The economical distribution of resources, such as transportation and inventory assets.

The screenshot below captures the results of the model.  


# 5. Analysis of Results:

The Genetic Algorithm reached an ideal solution with success. The optimal arrangement that minimizes transportation costs and meets product demand limitations is indicated by the assignment of products to locations ([1 4 2 0 3]), which represents the best solution.

The algorithm appears to have successfully identified a configuration where all products are sourced and transported at the lowest possible cost, as indicated by the best solution's least transportation cost. The task suggests:

Product 0 is assigned to Location 2

Product 1 is assigned to Location 5

Product 2 is assigned to Location 3

Product 3 is assigned to Location 1

Product 4 is assigned to Location 4

# 6. Conclusion:

The optimization methodology based on Genetic Algorithms has proven to be successful in tackling supply chain logistics problems. The products are assigned to places that best satisfy demand, save transportation expenses, and respect delivery deadlines. Using the Python Gurobi optimization has considerably increased the genetic algorithm model's success.