

# A Genetic Algorithm for Optimization Of Supply Chain Logistics Network

Rahma IDRIS

Second Year Engineering Student

Supervised by: Pr. Yassine Hachaichi

May 21, 2024

---

## Abstract

This project explores the optimization of supply chain logistics networks using genetic algorithms (GAs). The study begins by outlining the complexities inherent in modern supply chains, highlighting the need for efficient management strategies to mitigate challenges such as transportation costs, inventory control, and demand variability. Traditional linear programming approaches are discussed, revealing their limitations in addressing the intricate dynamics of supply chain networks.

Subsequently, genetic algorithms are introduced as a promising alternative for optimizing supply chain logistics networks. The principles of genetic algorithms, including selection, crossover, mutation, and fitness evaluation, are elucidated, emphasizing their suitability for handling complex optimization problems. A detailed implementation of the genetic algorithm in Python is provided, along with experimental results comparing its performance to traditional linear programming methods.

The findings demonstrate the efficacy of genetic algorithms in optimizing supply chain logistics networks, showcasing improvements in solution quality and convergence speed. The discussion delves into the strengths and limitations of the genetic algorithm approach, offering insights into its practical implications for real-world supply chain management.

In conclusion, this project underscores the potential of genetic algorithms as a valuable tool for enhancing the efficiency and resilience of supply chain logistics networks. The study contributes to the ongoing discourse on optimization techniques in supply chain management and identifies avenues for future research in this domain.

**Index Terms:** Logistics Optimization, Genetic Algorithms, Operations Research, Linear Programming, Genetic Algorithm Implementation, Python Programming.

---

## 1 Introduction

In today's interconnected global economy, efficient supply chain management is paramount for businesses seeking to gain a competitive edge. The intricate web of suppliers, manufacturers, distributors, and retailers that constitute modern supply chains presents a myriad of challenges, ranging from fluctuating consumer demands to unpredictable market dynamics. Effectively navigating these complexities requires sophisticated optimization techniques capable of streamlining operations, minimizing costs, and maximizing performance.

This research endeavors to address the optimization of supply chain logistics networks through the lens of genetic algorithms (GAs). Supply chain optimization is a multifaceted problem encompassing various factors such as transportation costs, inventory management, demand variability, and production scheduling. Traditional linear programming approaches have been widely employed to tackle these challenges, offering mathematical models to optimize resource allocation and flow within supply chains. However, as supply chains grow in complexity and scale, linear programming techniques may struggle to adequately capture the intricacies of real-world logistics networks.

Genetic algorithms, inspired by the principles of natural selection and evolution, offer a promising alternative for optimizing complex systems such as supply chains. Unlike traditional linear approaches, genetic algorithms operate through iterative processes of selection, crossover, mutation, and fitness evaluation, mimicking the process of natural selection to arrive at optimal solutions. By harnessing the power of genetic algorithms, supply chain managers can dynamically adapt to changing conditions,

optimize decision-making processes, and achieve superior performance outcomes.

This research aims to explore the efficacy of genetic algorithms in optimizing supply chain logistics networks. We will begin by outlining the challenges inherent in modern supply chains and the limitations of traditional linear programming approaches. Subsequently, we will introduce the principles of genetic algorithms and their application in the context of supply chain optimization. A detailed implementation of the genetic algorithm in Python will be provided, along with experimental results comparing its performance to traditional linear programming methods.

Through this research, we seek to demonstrate the effectiveness of genetic algorithms in enhancing the efficiency and resilience of supply chain logistics networks. By advancing our understanding of optimization techniques in supply chain management, we aim to provide valuable insights for practitioners and researchers alike, paving the way for improved decision-making processes and enhanced performance in real-world supply chain operations.

## 2 Supply Chain Logistics networks

*The problem entails optimizing supply chain logistics to minimize costs while meeting customer demand and operational constraints. The primary goal is to allocate resources efficiently, optimize transportation routes, and manage inventory effectively.*

### Objectives:

- **Cost Minimization:** Minimize total costs associated with production, transportation, and inventory holding while meet-

ing customer demand.

- Customer Demand: Ensure that customer demand is met at each production site.
- Resource Allocation: Optimize resource allocation, including production levels, transportation routes, and inventory levels, to achieve cost efficiency.
- Transportation Efficiency: Identify cost-effective transportation routes between production sites, warehouses, and distribution centers.
- Inventory Management: Implement inventory management strategies to minimize holding costs and prevent stockouts.

### Constraints:

- Production and Transportation Capacity: Limited capacity for production and transportation vehicles.
- Demand Fulfillment: Requirement to meet customer demand at each production site.
- Resource Limitations: Constraints on raw materials, labor, and equipment availability.
- Inventory Holding Costs: Costs associated with holding inventory, including storage and obsolescence.
- Operational Constraints: Factors like lead times and production schedules that impact logistics operations.

### Decision Variables:

- Production Levels: Quantity of goods produced at each site.
- Transportation Routes: Routes taken by vehicles for goods delivery.
- Inventory Levels: Quantity of goods held at each location.
- Resource Allocation: Distribution of resources across sites and operations.
- Order Fulfillment Strategies: Approaches for efficient order fulfillment, like batch processing and just-in-time inventory management.

Through addressing these objectives and constraints, the aim is to develop an optimized supply chain system that maximizes cost efficiency, customer satisfaction, and operational effectiveness.

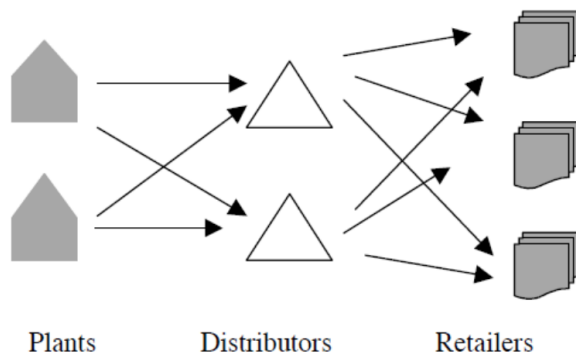


Figure 1. Two Stage Logistics Network Model

## 3 Problem Set

### Production Parameters:

- *pro*, *man*, *exp*, *log*, *imp*: These variables represent the number of products, manufacturing locations, export locations, logistics centers, and import locations respectively. In your code, *pro* = 6 indicates there are 6 different products, and *man* = 4 indicates there are 4 manufacturing locations.

### Distance Matrices:

- *distij*: Distance matrix between manufacturing locations (indexed by *i*) and export locations (indexed by *j*).
- *distjk*: Distance matrix between export locations (indexed by *j*) and logistics centers (indexed by *k*).
- *distli*: Distance matrix between logistics centers (indexed by *l*) and import locations (indexed by *i*).
- *distlj*: Distance matrix between import locations (indexed by *i*) and logistics centers (indexed by *j*).
- *distlk*: Distance matrix between logistics centers (indexed by *l*) and manufacturing locations (indexed by *k*).

### Inventory and Demand:

- *demand*: List representing the demand for each product.
- *lstock*: Matrix representing the initial stock levels at each location.
- *linv\_c*: Inventory holding cost per unit.
- *binv\_c*: Inventory holding cost per unit for backlog.

### Cost Parameters:

- *production\_cost\_per\_unit*: Production cost per unit.
- *tran\_rent1*, *tran\_rent2*: Transportation rental costs (though not utilized in the provided code).

### Genetic Algorithm Parameters:

- *max\_gen*: Maximum number of generations.
- *gen\_loc*: Generation location.
- *pop\_size*: Population size.

### Other Variables:

- *rand\_pop*: Randomly generated initial population.
- *children*: Holds the offspring generated in each generation.
- *tot\_avg*: List to store the total average cost across generations.

This initial setup provides the necessary data and parameters for running the genetic algorithm to optimize the production and inventory management problem based on the given constraints and demands.

## 4 Genetic algorithm

The genetic algorithm (GA) utilized in this project is a heuristic optimization technique inspired by the principles of natural selection and evolution. It operates on a population of candidate solutions, commonly referred to as individuals or chromosomes, and iteratively evolves these solutions over multiple generations to find near-optimal solutions to a given optimization problem.

At the heart of the genetic algorithm are several key components: selection, crossover, and mutation. In the initialization phase, a population of individuals is randomly generated to represent potential solutions to the optimization problem. Each individual in the population encodes a set of decision variables that define a candidate solution.

During the evolution process, individuals are evaluated based on their fitness, which represents how well they satisfy the objectives of the optimization problem. In the context of supply chain logistics, fitness could be measured as the total cost of production, transportation, and inventory management.

#### 4.1 Initialization:

In this phase, a population of individuals is randomly generated to represent potential solutions to the optimization problem. Each individual in the population encodes a set of decision variables that define a candidate solution. For example, in the context of supply chain logistics, each individual could represent a combination of production levels, transportation routes, and inventory management strategies.

#### 4.2 Selection:

Selection involves choosing individuals from the current population to serve as parents for the next generation. Individuals are selected probabilistically based on their fitness, with fitter individuals having a higher chance of being selected. Various selection techniques can be employed, such as roulette wheel selection, tournament selection, or rank-based selection.

#### 4.3 Crossover:

Crossover is applied to pairs of selected parents to produce offspring or children. During crossover, genetic material is exchanged between parents to create new combinations of decision variables. The crossover point is randomly selected, and genetic material beyond this point is swapped between parents to generate two offspring. This allows the genetic algorithm to explore different regions of the solution space and potentially discover better solutions through recombination.

#### 4.4 Mutation:

Mutation is a stochastic operator that introduces random changes to individual chromosomes. It promotes genetic diversity within the population by occasionally mutating certain decision variables. Mutation rates are typically low to prevent excessive disruption of good solutions but high enough to ensure exploration of the solution space. Mutation can occur at random positions within an individual's chromosome, with new values drawn from a specified range.

#### 4.5 Evaluation:

After the crossover and mutation operators are applied, the fitness of the resulting offspring is evaluated. Fitness represents how well an individual satisfies the objectives of the optimization problem. In the context of supply chain logistics, fitness could be measured as the total cost of production, transportation, and inventory management. The fitness evaluation determines which individuals are selected to proceed to the next generation.

#### 4.6 Replacement:

Once the offspring are evaluated, they are combined with a portion of the fittest individuals from the previous generation to form the next generation population. The population size remains constant throughout the evolution process, with individuals replaced based

on their fitness. This ensures that the genetic algorithm maintains diversity while focusing on promising regions of the solution space.

#### 4.7 Termination:

The evolution process continues for a specified number of generations or until a termination criterion is met. Common termination criteria include reaching a maximum number of generations, achieving a satisfactory solution, or stagnation of fitness improvement over successive generations. Upon termination, the best individual in the final population is selected as the solution to the optimization problem.

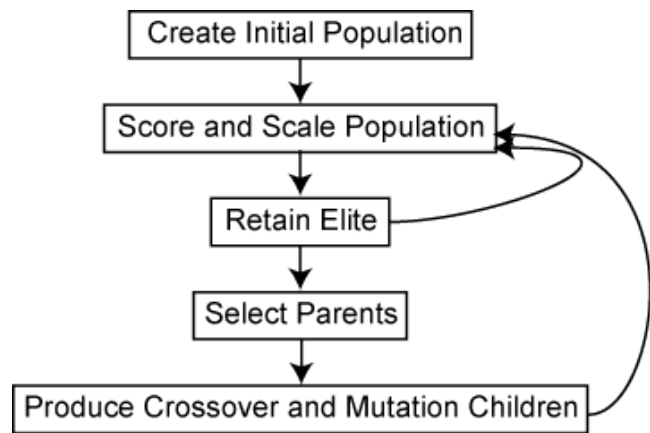


Figure 2. Genetic Algorithm

This graph shows the convergence of GA applied on our problem

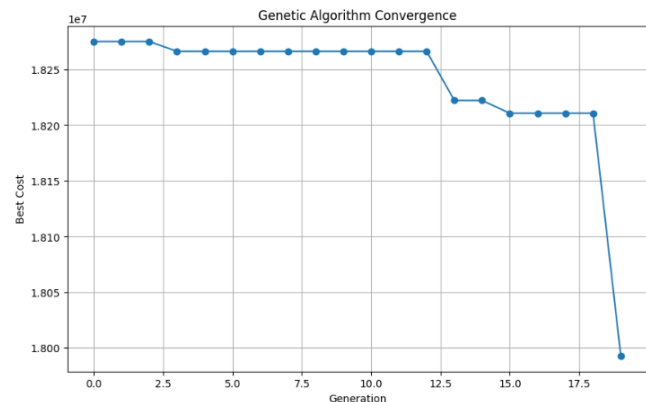


Figure 3. Convergence of GA

## 5 Linear Programming and GA

Linear programming (LP) offers another approach to optimization, particularly suited for problems with linear constraints and objectives. In the context of supply chain logistics, LP can be used to model and solve problems such as production planning, inventory management, and transportation optimization.

In an LP formulation, decision variables are defined to represent quantities or allocations, and linear constraints are imposed to

ensure that these variables satisfy operational requirements. The objective function, typically a linear combination of decision variables, is optimized subject to these constraints, aiming to minimize or maximize a certain criterion such as total cost or profit.

For example, in supply chain logistics, decision variables could represent production levels, inventory levels, and transportation quantities, while constraints could enforce capacity limitations, demand satisfaction, and resource availability. The objective function would then seek to minimize total costs, including production, transportation, and inventory holding costs.

The LP solver employs optimization algorithms such as the simplex method or interior-point methods to iteratively improve the solution until an optimal or near-optimal solution is found. LP solutions provide precise and deterministic results, making them suitable for decision-making in deterministic environments.

Comparing LP with genetic algorithms (GA) in the context of supply chain optimization, we can identify several key differences:

**Problem Complexity:** LP is well-suited for problems with linear constraints and objectives, while GA can handle more complex, non-linear, and combinatorial optimization problems. GA can explore a broader solution space and potentially find better solutions for highly nonlinear and non-convex problems present in supply chain logistics.

**Solution Quality:** LP provides precise and deterministic solutions, guaranteeing optimality under certain conditions. GA, on the other hand, provides heuristic solutions that may not always guarantee optimality but can find good solutions in a reasonable amount of time, especially for complex and non-deterministic problems.

**Computational Efficiency:** LP solvers can efficiently solve large-scale linear problems with thousands or even millions of variables and constraints. GA may require more computational resources, especially for large-scale or highly complex problems, due to the stochastic nature of the algorithm and the need to explore a vast solution space.

**Robustness to Uncertainty:** GA can handle uncertainty and variability in supply chain parameters and constraints more effectively than LP.

LP assumes deterministic inputs and may struggle to adapt to uncertain or dynamic environments without additional modeling techniques or modifications.

In summary, LP and GA offer different strengths and weaknesses for solving supply chain optimization problems. LP provides precise and deterministic solutions for linear problems but may struggle with complexity and uncertainty. GA, on the other hand, offers a flexible and robust approach for handling complex and nonlinear problems, making it suitable for uncertain and dynamic supply chain environments. The choice between LP and GA depends on the specific characteristics of the optimization problem, including its complexity, uncertainty, and objectives.

In this genetic algorithm iteration, we observed the cost gradually converging over generations, indicating optimization progress. The final solution achieved a certain cost, reflecting resource allocation and decisions made within constraints. However, linear programming failed to find a solution, preventing a comparison. Genetic algorithms excel in exploring diverse solutions for complex problems, albeit with higher computational demands. Linear programming offers precise solutions but struggles with complexity. The choice depends on problem characteristics and computational constraints.

```
Linear Programming did not find a solution
Genetic Algorithm Best Total Cost: 17993050
Comparison not possible as LP did not find a solution.
```

Figure 4. LP&GA

## 6 Conclusion

In conclusion, the genetic algorithm showcased its capability to iteratively improve solutions for the supply chain optimization problem, with the cost gradually converging over generations. Despite encountering challenges, such as computational complexity, the genetic algorithm successfully provided a feasible solution. On the other hand, the failure of linear programming to find a solution underscores its limitations in handling complex and nonlinear problems. Ultimately, the choice between these optimization techniques depends on the problem's characteristics, computational resources, and the desired level of solution precision. This comparative analysis sheds light on the strengths and limitations of each approach, providing valuable insights for decision-makers in supply chain management.