Optimization Strategy for End-to-End Supply Chain Planning

by

Javiera Paz Arancibia Bruce
Bachelor of Industrial Engineering, Universidad de Chile, 2015

and

María Fernanda Esparza Colin
Bachelor of Industrial Engineering, Tec de Monterrey, 2019

SUBMITTED TO THE PROGRAM IN SUPPLY CHAIN MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE IN SUPPLY CHAIN MANAGEMENT
AT THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

Signature of Author: ______________________________
Department of Supply Chain Management
May 16, 2025

Signature of Author: ______________________________
Department of Supply Chain Management
May 16, 2025

Certified by: ______________________________
Dr. María Jesús Saenz
Executive Director, MIT SCM master's Program
Capstone Advisor

Certified by: ______________________________
Dr. Jaime Macías
Postdoctoral Associate MIT SCM & MIT Digital Supply Chain Transformation Lab
Capstone Advisor

Accepted by: ______________________________
Prof. Yossi Sheffi
Director, Center for Transportation and Logistics
Elisha Gray II Professor of Engineering Systems
Professor, Civil and Environmental Engineering

Optimization Strategy for End-to-End Supply Chain Planning

by

Javiera Paz Arancibia Bruce

and

María Fernanda Esparza Colin

Submitted to the Program in Supply Chain Management
on May 16, 2025 in Partial Fulfillment of the
Requirements for the Degree of Master of Applied Science in Supply Chain Management

ABSTRACT

The sponsor company, a global consumer goods enterprise known for high-quality products and customer service, has been on a digital transformation journey over the past few years to make its supply chain more responsive. Currently, the company manages its end-to-end supply chain planning using Mixed Integer Linear Programming-based (MILP) software. This process takes approximately two hours, limiting the company's ability to rapidly update production and distribution plans in response to sudden changes in supply or demand.

Our capstone project proposes the use of metaheuristic models as an alternative to their existing planning software, with the goal of reducing planning time while minimizing total relevant costs. Specifically, we identified the conditions in which the company currently operates and developed a model configurator to optimize the end-to-end supply chain planning. The application of the configurator, based on Genetic Algorithm and Particle Swarm Optimization metaheuristics, was demonstrated across three representative demand scenarios and proved successful in reducing planning time by approximately 85%.

Additionally, the proposed solution maintained the same quality as the current solution of the company, achieving a 2% and 13% cost reduction in production and distribution respectively, while accounting for penalties related to unmet inventory targets. This improvement is significant, as it enables the company to become more responsive to internal and external changes, improving its ability to adapt quickly to dynamic supply and demand conditions.

Capstone Advisor: Dr. María Jesús Saenz
Title: Executive Director MIT SCM Master’s Program

Capstone Advisor: Dr. Jaime Macías
Title: Postdoctoral Associate, MIT SCM & MIT Digital Supply Chain Transformation Lab

ACKNOWLEDGMENTS

We would like to express our sincere appreciation to all those who contributed to the success of this project.

We are deeply grateful to our advisor, María Jesús Sáenz, for her guidance, expertise, and unwavering commitment throughout this project. Her strategic perspective and deep knowledge of supply chain innovation played a crucial role in shaping our approach.

We also wish to thank our co-advisor, Jaime Macías, for his valuable support, close collaboration, and practical insights, which greatly enriched the development and execution of our work. Their combined feedback and recommendations were fundamental in refining our methodology and elevating the quality of our outcomes.

We extend our sincere appreciation to our sponsor, Nathan, for his trust, engagement, and continuous support during the project. His openness to new ideas and consistent feedback were key in aligning our solutions with real-world needs.

Lastly, we thank our families and friends for their patience, encouragement, and unconditional support during this journey. Their belief in us was a constant source of motivation and resilience.

TABLE OF CONTENTS

1. INTRODUCTION

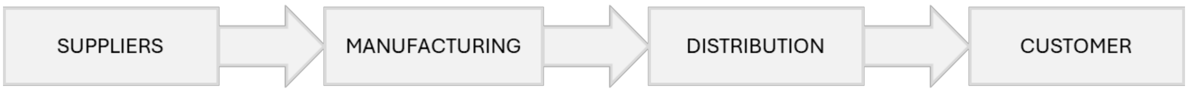
1.1 Background

Optimization and automation in supply chain processes are crucial, as they enable companies to quickly adapt to changes in demand, supply disruptions, or market fluctuations. This not only improves operational efficiency but also enhances flexibility and responsiveness to unexpected situations, making the supply chain more competitive and robust. A lack of automation and optimization in the supply chain leads to delays, errors, and higher operational costs. It also hampers efficient planning and the ability to quickly adapt to changes in demand or disruptions, impacting both the company's productivity and competitiveness.

1.2 Motivation

Our sponsor company, a global consumer goods enterprise known for its wide portfolio of categories such as personal care, household cleaning, and health products, is currently reviewing the optimization strategy for its end-to-end supply chain planning. The enterprise defines end-to-end supply chain planning as the process from purchasing raw materials, to manufacturing, inventory deployment, and finally to customer orders at downstream distribution centers, as is shown in Figure 1. The inventory deployment process refers to moving the product from the production facility to one of their 10 storage facilities.

Figure 1: End-to-End Supply Chain Planning Scope



The project will be centered on the Fabric Care division, with a particular focus on the Heavy-Duty Liquid (HDL) product line, which comprises approximately 600 Stock Keeping Units (SKUs). The manufacturing of the products is carried out in two production plants, A and B, with a total of 10 production lines. Each plant serves different markets, resulting in both plants producing similar SKUs.

The supply chain planning area is centralized for both production plants. At present, an automation process runs the manufacturing and product deployment planning program twice a day. The company utilizes specialized software, called OMP, to manage this operation. OMP is a provider of software solutions for supply chain planning and optimization. The data is transferred from SAP, a software used to manage business operations, to OMP, where Mixed Integer Programming (MIP) models are used for short-term planning at each plant. The objective of the MIP models is to determine the optimal quantity of each product to be produced and distributed to reduce inventory costs. To find the optimal solution, certain constraints are established, such as production line capacities, lead times, and Minimum Order Quantity (MOQ), among others.

According to the sponsor, the current system, OMP, is time-consuming and limits the overall responsiveness of the planning team, suggesting that it should operate faster. Furthermore, there are certain constraints that are not included in the planning model, for instance, storage capacity in the warehouses and raw materials availability, and these should be revised to assess their impact on the efficiency of the supply chain.

1.3 Problem Statement and Key Questions

The sponsor company has been working for years on the digital transformation of its supply chain. Aligned with their purpose of providing superior-quality products, they are working to create an optimization strategy and keep a responsive supply chain, capable of adjusting to supply, demand and market changes. Some of the important milestones on this path have been the implementation of an integrated business planning (SAP) and production optimization software (OMP). These systems were essential to achieve the company´s 98% service-level Key Performance Indicator (KPI), defined as the number of complete orders delivered on time divided by the total number of orders. However, the execution time required for production-to-distribution planning remains excessively long, thereby undermining the organization's ability to respond effectively to dynamic operational conditions.

Specifically, the manufacturing and deployment planning process relies on a single large program developed with linear constraints that determine what to manufacture, in what quantities, and where to store the products. To solve the planning problem within a maximum of 1.5 hours, as required by the sponsor, planners omit certain constraints from the program and "manually" apply them once the problem has been solved. If the program exceeds the allotted time, planners terminate the run prematurely and work with a suboptimal solution. Consequently, the company is seeking an alternative optimization strategy that allows it to reduce the planning execution time while ensuring costs minimization.

Reducing the planning execution time would allow planners to iterate and perform sensitivity analyses, which, in turn, could improve service, cash flow (inventory levels), and production costs. Nonetheless, the sponsor company main concern is improving their supply chain responsiveness. We are using the term supply chain responsiveness as the ability to respond to changes in supply and demand as defined by Christopher and Towill (2000). By selecting the best configuration of optimization models, the company would have the flexibility to quickly model different scenarios and make decisions in urgent situations. A model configurator can generate efficient production and distribution plans by leveraging the strengths of different metaheuristics. This directly addresses the execution time challenge, enabling faster and more responsive decision-making. In the context of the end-to-end planning process, the following questions will be addressed:

1. What are the key processes in the end-to-end supply chain planning? Which dependencies exist within those processes?
2. What are the variables and trade-offs that the current optimization model considers?
3. What supply disruptions and demand changes should the configuration consider to cover all mismatch scenarios?
4. What is the expected improvement in terms of cost and execution time resulting from the proposed model?

1.4 Project Goals and Expected Outcomes

The objective of the project is to develop a configurator of models to optimize the production and distribution planning process. It involves ensuring the management of overlapping objectives and determining the optimal limit of integration and constraints to guarantee solution quality and efficiency. To accomplish this, we will study the optimization model constraints and linkages for a particular business division in scope. We will also understand the tradeoffs between the quality of the model and running time to propose alternative solutions. To achieve the expected results, we will focus on a specific product category prioritized by the company. This scope includes production lines 2A, 3A, and 6A at Plant A, as well as lines 1B and 2B at Plant B. The proposed solution is designed to be scalable across the company, making it adaptable to other products in the future.

Creating an alternative model and comparing it to the current one requires a deep understanding of the processes, including variables such as product families, product types, and constraints like capacity and lead time. We will engage with stakeholders, particularly the Synchronized Planning Solutions team, and analyze historical data, including daily demand, capacity, and production rates for selected products. Subsequently, we will test various models to identify the optimal configuration that enhances supply chain responsiveness and service levels, while maintaining solution quality and reducing execution time. The findings will inform an assessment of current optimization practices and lead to actionable recommendations.

The main deliverable for the sponsor company will be a configuration to optimize the production and distribution planning process. This model will be able to select the best solution from a set of different methodologies, all implemented in Python. We will evaluate the quality of the solution by the total cost of the optimized production and distribution plans. The costs that the company is currently using are illustrative figures rather than precise costs drawn from their financial statements. We will be using the same costs to compare our solutions to theirs and estimating more accurate ones is out of scope. The demand forecast will also be provided by the company, and its calculation is out of scope.

Other deliverables to the company will include:

1. Diagnosis of the current planning strategy model in terms of supply chain responsiveness, service level, cost efficiency, and execution time.
2. Recommendations regarding the current planning strategy.

1.5 Plan of Work

The first step, following the problem definition, is to interview the stakeholders involved in the planning process to understand the existing variables and constraints. Additionally, we will explore the relationships between all stages of the supply chain and the optimization model. Next, we will formulate the objective function of our problem. After that, we will define a configuration of models that incorporates our objective function and constraints. We will then test and validate the configuration model, and the solution methods analyzed. Service level, cost to serve, inventory turnover, and execution time will serve as indicators to evaluate the program's performance and the quality of the solution. We will also iterate on the solution to achieve improvements. Based on the tests and results, we will propose the configuration of models that best fits different scenarios. Appendix A shows the iterative process to test and define and propose the configuration of models.

2. STATE OF THE PRACTICE

The primary problem that our capstone project will address is identifying the configuration for the supply chain optimization models for the planning of a consumer-goods company. We will determine the ideal optimization method to each scenario, and the necessary constraints to apply to guarantee the solution quality and efficiency, and as a result, increase the company's responsiveness and service level. To create the model, first we need to define the objective function. As we found more than one objective in our project, we defined our problem as multi-objective optimization.

Given the project objectives and challenges, we will review literature across several key areas. First, we will examine literature on responsiveness in supply chains and the key factors for improving it. Second, we will address the importance of planning the end-to-end supply chain. Finally, we will review different metaheuristic methods for solving the current multi-objective problem.

2.1 Supply Chain Responsiveness

Supply chain responsiveness is the capability of a supply chain to adjust to changes in demand, supply disruptions, or changes in the market environment. (Christopher & Towill, 2000). A responsive supply chain is characterized by its agility and flexibility, enabling companies to effectively meet customer demands even in uncertain conditions. Key factors that improve responsiveness include real-time monitoring, collaboration with suppliers and distributors, and agile inventory management, among other factors.

On the one hand, real-time monitoring allows companies to adjust their processes to reduce response times. This approach not only improves efficiency but also provides a competitive advantage, as companies can respond to changes in demand or supply chain disruptions (Suri, 1998).

On the other hand, developing strong collaboration with suppliers and distributors is essential for optimizing operational efficiency and responsiveness. Collaboration includes communication and information sharing between partners. By working closely with suppliers and distributors, companies can better anticipate and react to market fluctuations (Simchi-Levi & Kaminsky, 2007).

Finally, implementing agile inventory management practices is essential for managing fluctuations in demand and supply. Agile inventory management refers to maintaining flexible inventory strategies and having the ability to quickly adjust stock levels to support changes in the environment, such as demand fluctuations or a supplier's disruptions (Chopra & Meindl, 2015).

According to a study performed by Accenture, a customer-centric supply chain is key for reaching higher growth rates. After the COVID pandemic, 94% of consumer-packaged goods (CPG) companies have growth less than 3% per year. The common factor between those companies that have surpassed that growth threshold is a higher consumer value proposition through a connected consumer experience, price competitiveness, and trust and sustainability. As stated by Accenture, CPG companies require a tailored and responsive supply chain capable of reacting to the changes in different buying channels to achieve a higher value proposition. In this type of industry, management should focus on customer segmentation and building capabilities to deliver the right product portfolio at optimal cost. (Accenture, 2020).

Maintaining a responsive supply chain is a key goal that our sponsor company is pursuing through the optimization strategy. Responsiveness will enable the company to increase customer satisfaction, shorten lead times, and maintain a competitive advantage in changing markets. For the company, achieving a responsive supply chain requires updating their production and distribution plans multiple times a day or whenever there is a supply disruption. This means they need to shorten their current processes in order to meet this objective. (Chopra & Meindl, 2015).

2.2 End-to-end Supply Chain Planning

End-to-end supply chain planning has become critical over the last two decades. Globalization, new consumer channels, and wider SKU portfolios, among other factors, have exponentially increased complexity. At the same time, internal operation savings are reaching their limit since most of the retailers have already spent decades making their supply chains highly efficient. As a result, supply chain professionals are shifting to more collaborative approaches looking for value creation inside and outside their organizations. (Burnette & Dittmann, 2018).

The most resilient players of the consumer-goods industry during the COVID pandemic were the ones that had started implementing end-to-end planning. It is expected that over the next decade a high-performance planning function will help companies improve their top and bottom lines (increase revenue and reduce costs). (Ghandour, et al., 2021).

Regarding the top line, companies can take advantage of new revenue streams (for example, ecommerce), and acquire competitive advantage from resilience in the case of supply chain disruption during global crises (i.e., the COVID pandemic). Regarding the bottom line, production and logistics costs have been increasing because of changing customer expectations for product portfolios' variety, sustainable sourcing, reduced delivery time, and other consumer trends. To ensure optimal decisions, organizations require cutting-edge algorithms and accurate information.

The principles that enable a successful implementation of end-to-end planning strategy are (Ghandour, et al., 2021):

1. Cross-functional integration to take the decisions that create value.
2. Short planning cycles to respond to changes in demand.
3. Use of advanced analytics to improve forecasts and planning.
4. Automation of standard tasks to allow planners to focus on decision making.

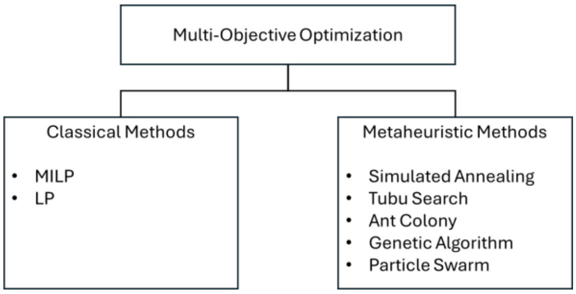
Inside a company there are multiple objectives to satisfy, such as inventory reduction, demand fulfillment, and efficiency on production lines, among other objectives. Sometimes this variety of objectives can overlap and generate contradictions. A typical example is the relationship between minimization of total cost in the entire supply chain and the maximization of the service level. Increasing the service level increases customer satisfaction, however, this tends to increase inventory levels or incur extra costs. The keys to balancing these overlapping constraints are cross-functional integration and the use of advanced analytics. (Ghandour, et al., 2021)

2.3 Techniques for Multi-objective Optimization Problems

In multi-objective optimization problems, there is more than one objective function to be satisfied simultaneously. In some cases, the objectives overlap, leading to conflicts between them. (Trisna, et al., 2016). In multi-objective optimization problems, there is no single global solution. Instead, a set of solutions can be found that form Pareto optimal solutions. This means that there is a set of solutions (trade-offs) among the different defined objectives (Deb, 2001).

According to Donoso and Fabregat (2007) we can divide multi-objective optimization problems into two categories: classical (or traditional) methods and metaheuristic methods. Figure 3 shows the classification mentioned. On the one hand, classical methods convert multi-objective problems into single-objective problems by aggregating objective functions, and these problems are often solved using techniques like mixed-integer linear programming (MILP). On the other hand, metaheuristic methods can guide other heuristic methods or algorithms in the search for feasible solutions of the optimal value or the set of optimal values.

Figure 2: Classification of Multi-objective Optimization



Mixed Integer Linear Programming tools are commonly used to solve supply chain network design problems. These tools can obtain an optimal solution within a wide range of linear constraints. However, MILP is computationally expensive and time-consuming. For that reason, different alternatives have been explored, such as metaheuristic approaches to solve complex supply chain models.

Different metaheuristic methods can solve multi-objective optimization problems. We will focus on five relevant approaches: Simulated Annealing, Tabu Search, Ant Colony, Genetic Algorithm and Particle Swarm. These methods can find heuristic solutions to complex problems in an efficient way. For the full list of researched metaheuristic optimization methods refer to Appendix B. Table 1 shows a brief comparison of the techniques that will be explored.

Table 1: Metaheuristic Methods Comparison Chart

Feature	Tabu Search (TS)	Simulated Annealing (SA)	Ant Colony Optimization (ACO)	Genetic Algorithm (GA)	Particle Swarm Optimization (PSO)
Inspiration	Problem-solving strategies	Annealing process in metallurgy	Behavior of ants in finding optimal paths using pheromones	Natural evolution process	Movement of flocks of birds or schools of fish
Mechanism	Explores neighborhoods while avoiding revisiting past states using a tabu list	Accepts changes that increase the objective function with a decreasing probability	Ants deposit pheromones to mark paths, influencing others	Combines recombination, mutation, and selection to generate solutions	Particles move through solution space based on personal and group behaviors

Key Parameter	- Tabu list size - Number of iterations - Neighborhood size	- Initial temperature - Annealing rate	- Pheromone evaporation rate -Concentration of pheromone - Ant colony size	- Population size - Crossover probability - Mutation probability	- Number of particles - Velocity of each particle - Social influence parameter
Key Strengths	- Avoids local optima effectively	- Avoids being trapped at local optimal	- Avoids convergence of suboptimal solutions	- Is effective in large and complex solution spaces	- Balance between local and global optima

2.3.1 Genetic Algorithm

The Genetic Algorithm (GA) is an attempt to simulate the natural evolution process where attributes are modified by the exchange and combination of chromosomes during breeding. The two main principles are that complicated structures can be represented by simple bit strings, and that those strings can be improved by simple transformations.

Like Simulated Annealing, the GA only uses the objective function information, not its derivatives, and it uses probabilistic transition rules. However, the GA uses an encoding of its control variables instead of the variables themselves, and searches from one population of solutions to another. The GA consists of creating an initial "population" of solutions encoded in binary bit strings. When encoding continuous control variables, the accuracy will depend on the length of the bit strings leading to a trade-off between precision and running time. Different from other optimization routines, the process of generating a new solution consists of three activities: selection, recombination (or breeding), and mutation. (Parks, G., & Sepulchre, R., 2020).

2.3.2 Particle Swarm Optimization

The Particle Swarm methodology is a heuristic optimization algorithm inspired by the social behavior of groups of animals like fish schooling, birds flocking, or honeybees flying (Gad, 2022). Everyone within a swarm has their own simple capacities to find a solution. However, they perform in a collaborative way, interacting among themselves to find the best solution. The interactions could be direct (visual or auditive) or indirect (reacting to changes in the environment).

The methodology considers that everyone is a particle. Each individual is a potential solution that is viewed as a particle with a specific velocity moving through a defined space. Each solution combines data of historical best location, current location and data of other individuals of the swarm to define the next movement. The next iteration occurs after all particles have been moved. Particles adjust their velocity and position, converging towards the optimal solution as the swarm's collective knowledge improves. (Dorigo, M., Maniezzo, V., & Colorni, A., 1996).

The primary objective of the particle swarm methodology is to create a balance between each individual particle and the swarm (local and global). It needs a few adjustable parameters since it is computationally efficient and applicable to a wide range of optimization problems.

3. METHODOLOGY

In this chapter, we summarize the approach to developing a model configurator to optimize production and distribution planning, that balances the trade-off between production capacity, inventory relocation, and service level, considering that the execution time and quality of the solution are key factors. In developing our model, we adapted methodologies and incorporated considerations regarding different metaheuristic methods, as described in Section 2.3.

3.1 Project Framework

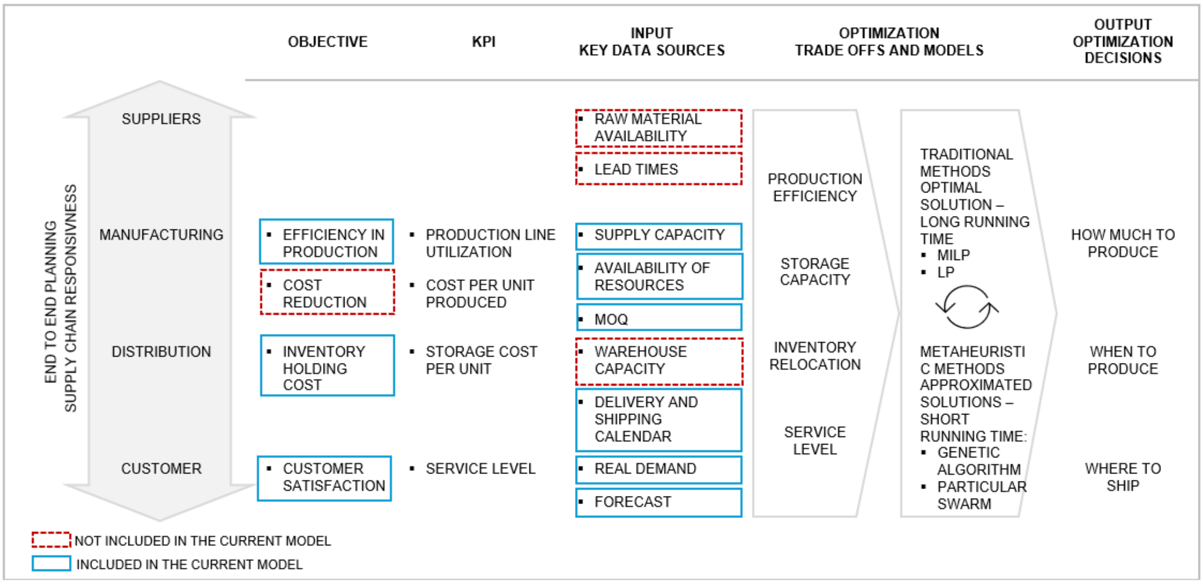
The sponsor company is recognized by its customers for its high-quality products. Therefore, having a robust and highly responsive supply chain is key for their business strategy. Creating a configuration for the supply chain optimization models can help to reach this goal. According to our literature review, agile inventory practices and collaboration are essential in responsive supply chains. However, as mentioned in Section 2.2, flexibility in inventory practices, responsiveness, and overall service level are opposing objectives to a cost-efficient supply chain. Those different goals can be managed through an end-to-end planning approach.

The sponsor company is currently using MIP-based software, which is highly time-consuming and restricts the constraints that can be placed on the model. Metaheuristics are another alternative to optimization problems, but unlike linear programming, metaheuristics do not look for exact solutions but sufficiently good ones. These techniques can be an appropriate option since they require less time and computational power. Based on the literature reviewed, we have concluded that the most appropriate techniques are the Genetic Algorithm and the Particle Swarm.

We selected Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) based on the nature of the production and distribution planning problem and the need for faster, scalable solutions compared to traditional Mixed-Integer Programming (MIP) approaches. Genetic Algorithm was chosen due to its ability to explore large and complex solution spaces efficiently, handle discrete and nonlinear decision variables, such as production quantities, scheduling, and plant assignments. Additionally, GA avoids getting stuck in local optima through mechanisms like crossover and mutation. GA is easily adaptable to custom constraints (e.g., minimum order quantities, line priorities), which are difficult to incorporate in traditional solvers. On the other hand, Particle Swarm Optimization was selected because it shows rapid convergence in continuous or near-continuous decision spaces, such as production levels or distribution flows. It requires fewer hyperparameters and tends to be simpler to implement and tune. PS performs well in scenarios with smooth cost surfaces, such as inventory holding, transportation, or penalty costs. By combining both approaches, we are able to leverage the strengths of GA in combinatorial aspects of the problem, and the speed of PSO in continuous domains, achieving feasible and high-quality solutions in significantly less time than MIP.

Figure 4 illustrates our project framework, which provides an organized scheme for addressing the problem. As defined by the sponsor company, the end-to-end supply chain covers four main stages: suppliers, manufacturing, distribution, and customer. The right coordination and balance between these stages will enable responsiveness in the supply chain, a key objective for the enterprise. Each process has its objectives. For instance, cost reduction or efficiency in production lines are fundamental goals for the stage of manufacturing. In some cases, different objectives can overlap, generating conflicts among themselves.

Figure 3: Project Framework



Some of the constraints showed in figure 4 are included in the current optimization model, such as supply capacity, over inventory cost, and demand. However, there are some constraints that are not included in the model and must be addressed to achieve optimized results across the supply chain.

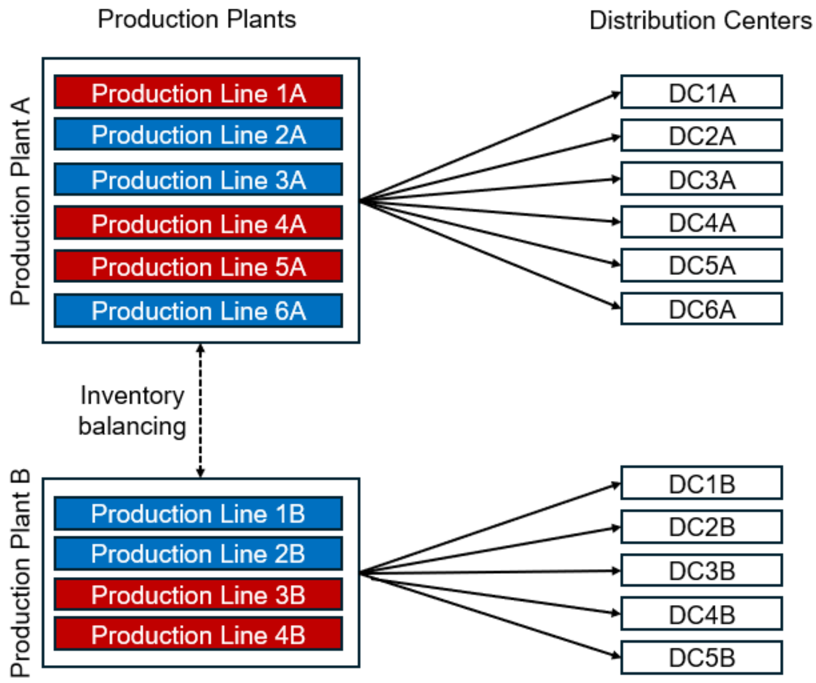
After the variables and constraints are defined in the model, it is possible to formulate the optimization problem and determine how to solve it. We will explore the current methodology (traditional methods), and metaheuristic approaches to find a better solution in terms of efficiency and quality. Finally, the output of the model recommends which SKUs and how much to produce, when to produce and where to ship them to achieve a balance between the defined objectives and tradeoffs.

3.2 Production-Distribution Network Structure

The first part of the data analysis phase consisted of understanding the supply chain structure of the sponsor company, through interviewing stakeholders, specifically, the Synchronized Planning Solutions team. The key features to identify were the processes, objectives, constraints, and variables, and which of them were already included in the company's optimization model. Sensitive data was masked to protect confidentiality.

To build our model, we received data related to the demand for each SKU-distribution center pair, safety stock levels per SKU-distribution center, production rates for each production line and SKU, the bill of materials for each SKU, lead times between production plants and distribution centers, and current stock levels across distribution centers. Figure 5 illustrates the structure of the production and distribution network. There are two production plants: Plant A and Plant B. Plant A consists of six production lines (Line 1A to Line 6A) and Plant B has four production lines (Line 1B to Line 4B). Each production Plant can serve a defined group of distribution centers. Plant A supplies products to DC1A through DC6A and Plant B supplies products to DC1B through DC5B. Additionally, there is an inventory balancing mechanism between the two plants, allowing them to transfer stock to optimize availability across the network. The red and blue colors differentiate between the two types of products that the plants can produce. Each group of lines is exclusive for each type of product. It means that production line 1A, 4A, and 5A in Plant A and production lines 3B and 4B in Plant B can produce the same type of products.

Figure 4: Production Network Structure and Distribution Flow



3.3 Problem Scope Definition

Since each SKU type is produced on a specific set of production lines, managing production allocation across both product types is unnecessary. To narrow the model's scope and reduce computational complexity, we selected one SKU type while ensuring that the chosen subset remained mutually exclusive and collectively exhaustive. Both product types have a similar volume share; however, in agreement with the company, we decided to focus on the company's key product: Product Type Blue and deliver a functional prototype that can be scaled to both product types. Figure 6 shows the daily demand for both product types over the next 90 days. Table 2 presents a statistical analysis of the behavior of the two categories. It shows that the selected product type accounts for 45% of the total demand.

Figure 5: Forecast Demand (Units) 90 Days – Type of Product

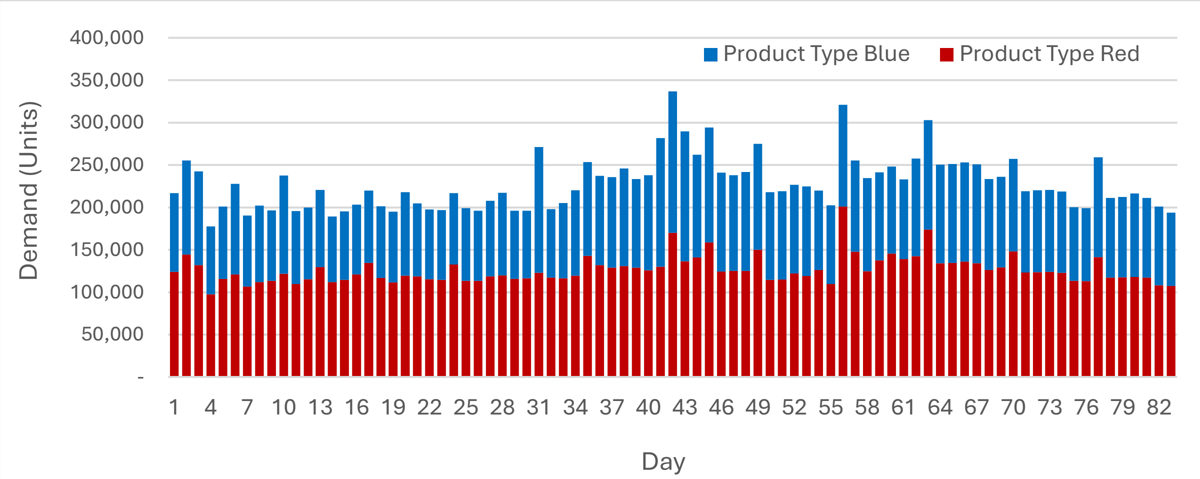


Table 2: Demand Units Summary Statistics

	Daily Demand					
Type of Product	Mean (units)	Std. Dev. (units)	CV (%)	Max (units)	Min (units)	Share (%)
Red	126,314	15,998	0.13	200,979	97,403	55.4%
Blue	101,805	18,228	0.18	167,108	77,738	44.6%
Total	228,119	31,075	0.14	336,954	177,692	100.0%

3.4 Model Specification and Development

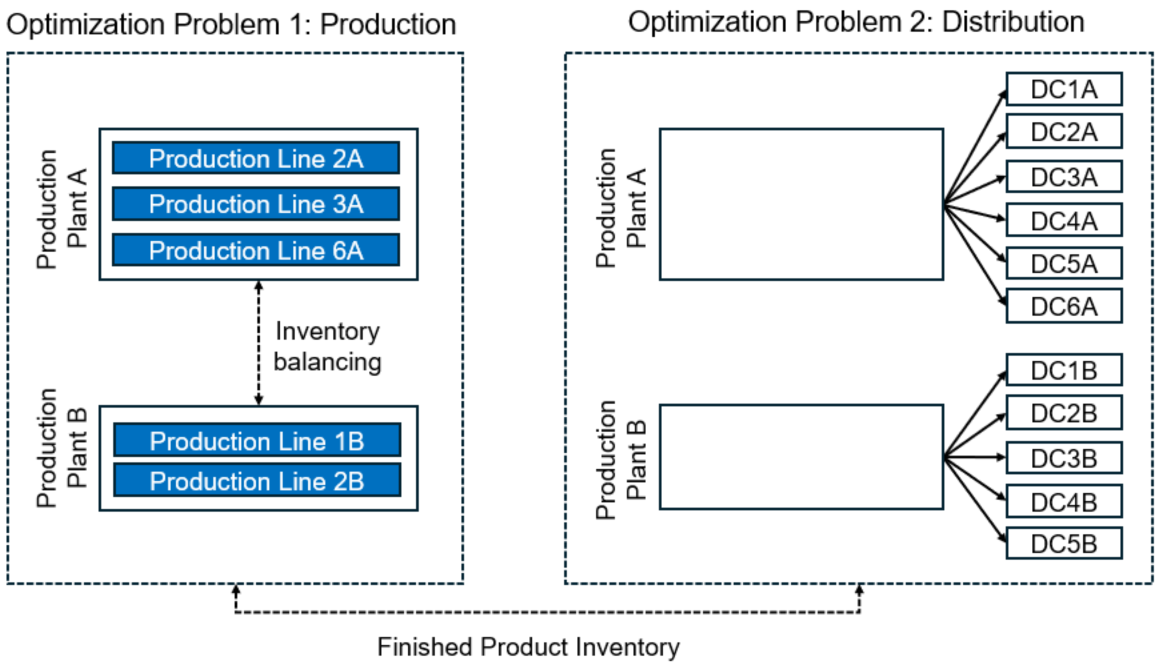
We define the use of a "Hybrid decomposition optimization strategy" to solve a complex problem by breaking it down into smaller, more manageable parts and optimizing each component using different methods. There are two steps involved in this strategy: a) Decomposition approach and b) Three-phase hybrid approach.

To build a decomposition approach, it is first necessary to preprocess the data by aggregating the requirements from each distribution center and converting them into production plant requirements. This involves checking future demand and inventory levels to define the specific requirements for each production line. After this process, we can proceed with the decomposition approach.

3.4.1 Decomposition Approach

The first step in the strategy involves designing a decomposition approach that decouples elements with no dependencies. The decomposition focuses on distinct components within the supply chain. We separated the production and distribution problem as shown in Figure 6.

Figure 6: Decomposition Approach



The first optimization problem, Production, aims to determine the appropriate SKUs and quantities to produce, based on the initial production requirements for each plant. This step defines what to produce and in what amount, while considering capacity constraints at the production facilities. The second optimization problem, Distribution, focuses on determining where to ship the available inventory, given that the SKUs and their quantities have already been established in the first problem. This decomposition ensures that each segment can be optimized individually while maintaining coherence with the overall supply chain objectives.

3.4.2 Three-Phase Hybrid Approach

After decomposition, a three-phase hybrid optimization process is implemented. Each phase aims to determine whether the optimization problems can be solved within a single production plant, meaning without sharing resources between plants. This would ensure that each optimization problem remains small and can be solved more easily and quickly. If the capacity is not sufficient, a shared-resource, more complex problem must be solved instead. A configuration model is created using two different metaheuristic models (Genetic Algorithm and Particle Swarm) to explore the search space efficiently.

- Genetic Algorithm (GA): Provides robust exploration using selection, crossover, and mutation. Includes parameters such as population size, chromosome representation, gen representation, and initial population.
- Particle Swarm Optimization (PSO): Features coordinated search and rapid convergence. Requires low computational cost. Includes parameters such as population size, number of iterations, inertia weight, cognitive parameter, and social parameter.

The objective function and associated constraints are formulated for both optimization problems, production and distribution, and are solved using the configuration approach proposed.

3.4.2.1 Production Planning Optimization

For the Production problem, we define the objective function and the constraints. Table 3 presents the notation adopted throughout the formulation, Table 4 details the model parameters, and Table 5 defines the decision variables involved in the optimization process.

Table 3: Notation for the Formulation of the Production Problem

Symbol	Description	Unit
T	Planning horizon = {1,...,14}	days
t	Time index	NA
P	Set of SKUs = {1,...., P}	SKU
p	SKU index	NA
L	Production line	NA

We define a 14-day planning horizon, as this is the timeframe currently used by the company for optimization purposes.

Table 4: Parameters of the Production Problem

Symbol	Description	Unit
C_{IOT}	Cost of holding inventory over the target	\$
C_{IUT}	Cost of holding inventory under the target	\$
C_{IUM}	Cost of holding inventory under the minimum	\$
$I_{OT,p,t}$	Units of inventory over the target of product p in period t	Un
$I_{UT,p,t}$	Units of inventory under the target of product p in period t	Un
$I_{UM,p,t}$	Units of inventory under the minimum of product p in period t	Un

I_p^t	Target inventory of product p	Un
I_p^m	Min inventory of product p	Un
$C_{B,p,t}$	Cost of stockout	$
$B_{p,t}$	Units of inventory stockout of product p in period t	Un
$I_{p,t}$	Units of inventory of product p in period t	Un
r_p	Production Rate of product p	Un/ Hr
$D_{p,t}$	Units demand of product p in period t	Un
H_t	Maximum hours in period t	Hr

Table 1: Decision Variables for the Production Problem

Symbol	Description	Unit
$x_{p,t,L}$	Units to manufacture of product p during period t in production line L	Un

Objective Function

The objective function in the first optimization problem is to satisfy the demand to achieve the target service level while maintaining target inventory levels. To define this objective function, we first identify what we want to minimize and the components that determine which SKUs to produce and in what quantities. These components are:

- Cost of holding inventory over the target: Calculated as the sum product of the inventory over the target $I_{OTp,t}$ of all SKUs and the cost c_{IOT}, aggregated for all periods.

$$\sum_p^P \sum_t^T C_{IOT} * \ I_{OTp,t} \ \forall \, p \, \epsilon \, P; \ t \in T \tag{Eq 2}$$

- Cost of holding inventory under the target: Calculated as the sum product of the inventory under the target $I_{UTp,t}$ of all SKUs and the cost c_{IUT}, aggregated for all periods.

$$\sum_p^P \sum_t^T C_{IUT} * \ I_{UTp,t} \ \forall \, p \, \epsilon \, P; \ t \in T \tag{Eq 3}$$

- Cost of holding inventory under the minimum: Calculated as the sum product of the inventory under the minimum $I_{UMp,t}$ of all SKUs and the cost c_{IUM}, aggregated for all periods. In this case, the company assumes that the minimum inventory is 0.9 * target inventory.

$$\sum_p^P \sum_t^T C_{IUM} * \ I_{UMp,t} \ \forall \, p \, \epsilon \, P; \ t \in T \tag{Eq 4}$$

- Cost of stockout: It is calculated as the sum product of the inventory stockout $B_{p,t}$ of all SKUs and the cost C_B, aggregated for all periods.

$$\sum_p^P \sum_t^T C_B * B_{p,t} \ \forall\, p \in P;\ t \in T \qquad \text{(Eq 5)}$$

Finally, our objective function Minimizes cost as a result of holding inventory over or under targets and stockout penalties.

$$Min\ \sum_p^P \sum_t^T C_{IOT} * I_{OTp,t} + \sum_p^P \sum_t^T C_{IUT} * I_{UTp,t} + \sum_p^P \sum_t^T C_{IUM} * I_{UMp,t} + \sum_p^P \sum_t^T C_B * B_{p,t} \qquad \text{(Eq 6)}$$

Constraints

- Production Capacity constraint: Limits total production per period based on capacity of each production line.

$$\sum_p^P x_{p,t} * r_p - H_t \leq 0,\ \forall t \in T \qquad \text{(Eq 7)}$$

- Inventory Balance: Tracks inventory changes across periods.

$$I_{p,t} - I_{p,t-1} + x_{p,t,L} + D_{p,t} + B_{p,t-1} - B_{p,t} = 0,\ \forall t, \forall p \qquad \text{(Eq 8)}$$

- Inventory Target deviation:

$$I_{p,t} - I_p^t \leq I_{OT,p,t},\ I_{OT,p,t} \geq 0,\ \forall t, \forall p \qquad \text{(Eq 9)}$$

$$I_p^t - I_{p,t} \leq I_{UT,p,t},\ I_{UT,p,t} \geq 0,\ \forall t, \forall p \qquad \text{(Eq 10)}$$

- Inventory Minimum deviation:

$$I_{p,t} + I_{UM,p,t} \geq I_p^m,\ I_{UM,p,t} \geq 0,\ \forall t, \forall p \qquad \text{(Eq 11)}$$

3.4.2.2 Distribution Planning Optimization

The objective function in the second optimization problem is to satisfy the demand in each distribution center to achieve the target service level while maintaining target inventory levels. For the Distribution problem we define another objective function and constraints. Table 6 presents the notation adopted throughout the formulation, Table 7 details the model parameters, and Table 8 defines the decision variables involved in the optimization process.

Table 6: Notation for the Formulation of the Distribution Problem

Symbol	Description	Unit
J	Set of Distribution centers = {1,...,10}	NA
j	Distribution center index	NA
P	Set of SKUs = {1,...., P}	SKU
p	SKU index	NA

Table 7: Parameters of the Distribution Problem

Symbol	Description	Unit
$C_{p,j}$	Cost of sending inventory of SKU p to Distribution Center j	$
$D_{p,j,t}$	Units demand of product p in DC j in period t	Un
$B_{p,j,t}$	Units Stockout of product p in DC j in period t	Un
$I_{p,j,t}$	Units of inventory of product p in DC j in period t	Un

Table 8: Decision Variables for the Distribution Problem

Symbol	Description	Unit
$x_{p,j,t}$	Units to send of product p to DC j in period t	Un

Objective Function

To define the objective function in the second problem, we must consider the transportation cost of sending a product p to Distribution Center j in period t and total units to send:

$$Min \sum_{p}^{P} C_{pj} * x_{pjt} \ \forall \ t \in T \qquad \text{(Eq 12)}$$

Constraints

- Demand constraint: Satisfy the requirements of each distribution center.

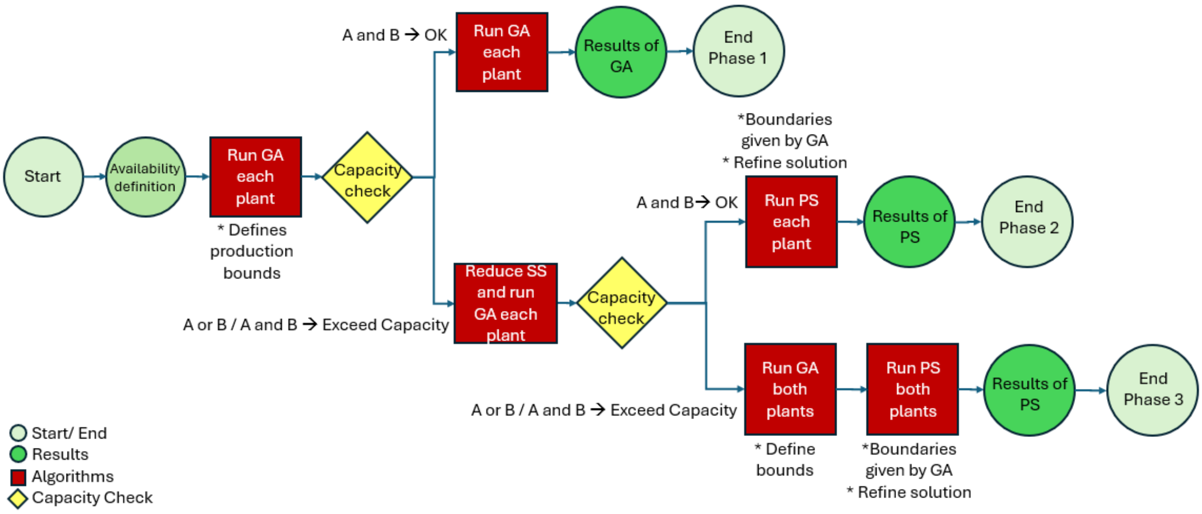
$$D_{pjt} - B_{pjt-1} - x_{pjt} \leq 0, \ \forall t \ \in T \qquad \text{(Eq 13)}$$

- Inventory constraint: Satisfy the inventory balance constraint in each distribution center.

$$x_{pjt} + B_{pjt-1} - I_{pjt} \leq 0, \ \forall t \ \in T \qquad \text{(Eq 14)}$$

With a clear understanding of the two optimization problems to be solved, we can apply our model configurator using the Three-Phase Hybrid Approach. This means that both optimization problems are solved when applying each metaheuristic algorithm, following the structure presented in Figure 7 as further described below.

Figure 7: Three-Phase Hybrid Approach: Configuration of Models



The three-phase hybrid approach begins by defining the availability of each production line. To define this availability, we can identify expected and unexpected changes in the supply and production as shown in Table 9.

Table 9: Causes for Changes in the Availability of Resources

	Expected	Unexpected
Production	Preventive maintenance Programed infrastructure upgrades	Corrective maintenance in production lines Quality defects in production
Supply	None	Disruptions in the supply chain of raw materials vendors Transit or weather incidents that constraint delivery

Additionally, it is important to account for increases in demand and to distinguish between internal factors (such as operations and promotions) and external ones.

After this definition, the process begins by independently running a Genetic Algorithm (GA) for each plant. The objective of this step is to quickly assess each plant's capacity. It also helps define the bounds for the optimization in the subsequent steps.

Once the GA is executed, a capacity check is performed. If both plants, plant A and B, operate within their capacity limits, the process proceeds to a second GA run with updated parameters of population and generation size, to reach a more refined solution. Still optimizing each plant individually. The results from this GA run are then collected, which provides the optimization results for the first phase.

If during the first capacity check either plant A or B exceed its respective capacity limits (or both do), the system proceeds to reduce the safety stock (SS) and reruns the GA for each plant individually. A second capacity check follows. If both plants remain within capacity, the Particle Swarm Optimization (PS) algorithm is run separately for each plant using the boundaries determined by the GA. The objective of this step is to refine the solution given by the GA. The results from this PS run are then collected, which provides the optimization results for the second phase.

Finally, If the capacities are still exceeded, a joint Genetic Algorithm is run for both plants simultaneously using newly defined bounds, followed by a joint Particle Swarm run using the boundaries (limited search space) provided by the GA. This final stage produces optimized results, ensuring that both plants meet capacity requirements while sharing resources. This is the end of the third phase.

This methodology ensures a coordinated, computationally efficient approach to supply chain optimization by leveraging the strengths of metaheuristics for initial solution exploration and solution refinement.

4. RESULTS AND DISCUSSION

After developing the optimization model described in Section 3, we adjusted the parametrization of the metaheuristic models to identify the right balance between solution quality and execution time. To assess these trade-offs, we designed a set of illustrative scenarios that demonstrate the model's logic, adaptability, and performance under diverse supply and demand conditions. Each scenario reflects a different degree of alignment, or misalignment, between supply and demand:

- Scenario 1 - Baseline capacity match: Each plant has enough capacity to meet its own demand and maintain its target inventory levels.
- Scenario 2 - Capacity meets demand but falls below target inventory: Each plant has enough capacity to meet its own demand, but not to maintain the target inventory levels for every SKU.
- Scenario 3 – Inventory transfer between plants due to capacity shortfall: Demand cannot be satisfied by one of the plants, even by using the safety stock. Transferring inventory between plants is necessary.

Although the number of SKUs or units that fall below target inventory or stockout may vary, these three scenarios capture the primary demand fulfillment conditions. In practice, Scenarios 1 and 2 are the most common, but including Scenario 3 ensures that the model is tested under every possible condition and allows us to measure execution time for each case.

This section is structured as follows. Subsection 4.1 describes the testing and parameter tuning of the models. Then, Subsections 4.1.1 to 4.1.3 cover scenarios 1 to 3, respectively. Finally, Subsection 4.2 provides a summary and comparison of the proposed models against the company's current solution in Section 4.2.

4.1 Model Testing and Parameter Tuning

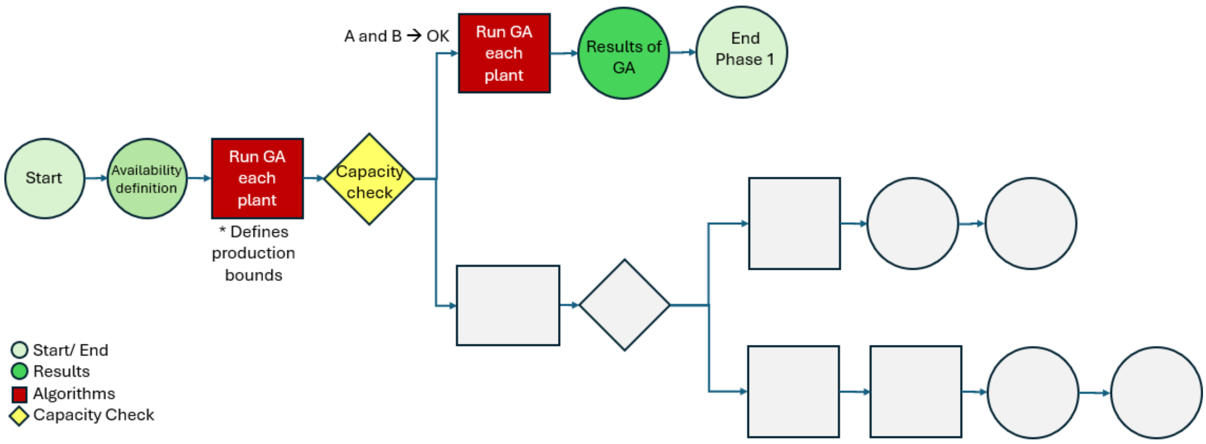
As we explained in Section 2.3, a limitation of metaheuristic techniques is that they do not necessarily obtain the optimal solution, although they can obtain a solution close to it in less time than MILP-based optimizers. The trade-off between the quality of the solution (in this case, cost reduction), and the time to run the program is determined by the parameters used by the metaheuristic. In the case of Genetic Algorithm, the parameters used are population size, number of generations, crossover rate, and mutation rate. Similarly, the Particle Swarm methodology uses population size, number of iterations, inertia rate, cognitive parameter, and social parameter.

In this section, we analyze the performance of each step in the configuration model by examining the parameters and results for each step and algorithm. To do this, we create different scenarios to test the three phases of the "Three-Phase Hybrid Approach".

4.1.1 Scenario 1: Baseline Capacity Match

We analyzed the scenario in which all the requirements at each plant can be satisfied without using the safety stock. Scenario 1 follows the path shown in colors in Figure 8.

Figure 8: Path in the Configuration of Models for Scenario 1



We analyzed Plant A. The initial parameters for this scenario are shown in Table 10.

Table 10: Parameters Scenario 1

Plant	N° of SKUs	Avg Daily Demand (Units)	Initial Inventory (Units)	Inventory (Days)
A	200	32,038	508,702	15.8

As represented in Figure 8, once we have defined the availability of resources, the Genetic Algorithm is performed. This Algorithm is less restrictive and does not consider MOQ constraints. This is because we need to perform a quick evaluation of the plant's capacity. At the beginning, the GA algorithm is executed using the parameters shown in Table 11. The initial parameters were selected following the recommendations given by Boyabatli and Sabuncuoglu (2004).

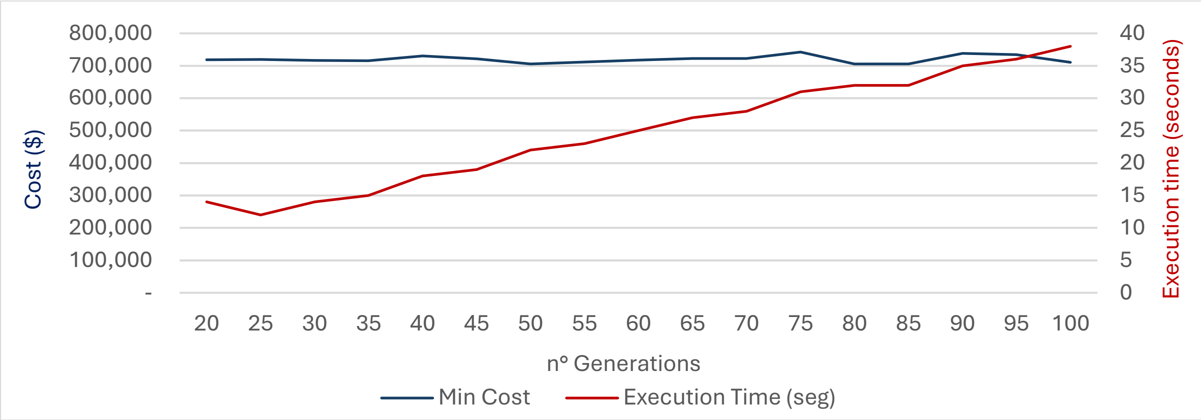
Table 11: Parameters Genetic Algorithm

Parameter	Value	Description
Population Size	20	Number of candidate solutions per generation
Generations	50	Number of iterations the algorithm runs
Crossover Rate	0.5	Probability of combining two solutions to create offspring
Mutation Rate	0.05	Probability of randomly altering a solution to maintain diversity

Then we conducted a sensitivity analysis, changing the number of generations in the algorithm. The number of generations is a key parameter in Genetic Algorithms, as it determines how many iterations the evolutionary process will undergo. Each generation represents a cycle of selection, crossover, and mutation, allowing the algorithm to progressively improve the quality of the solutions. A higher number of generations generally enables better convergence toward optimal or near-optimal solutions, but it also increases computational time. Therefore, choosing an appropriate number of generations involves balancing solution quality and computational efficiency. This step is not meant to be repeated each time the configurator is used, but only initially to set up the GA.

Figure 10 shows the change in the solution (cost) and the execution time while increasing the number of generations in the model. As observed, the improvement in the optimal solution is moderate throughout the analysis, with no more than a 5% difference between the best and worst solutions. Therefore, we decided to set the number of generations to 50 as was recommended in Boyabatli and Sabuncuoglu (2004). This value is a reasonable time for our model (22 seconds) and ensures a good-quality solution while allowing us to efficiently proceed to the next step in the configuration model. Appendix C shows how the algorithm can find the optimal solution for the production plan in Generation 28.

Figure 9: Impact of the Number of Generations on Cost Optimization and Execution Time in the Genetic Algorithm



This first run allows us to perform an initial capacity check. It helps us determine whether the plant has enough capacity to meet all requirements, measured as a percentage of production plant utilization. The first scenario occurs when both production plants can satisfy all requirements, and the optimal solution is obtained in the first phase of the configuration model. The result of this first run give us the utilization of each production Plant. For the analysis of Plant A, the results after running the first capacity check in Scenario 1 are shown in Table 12.

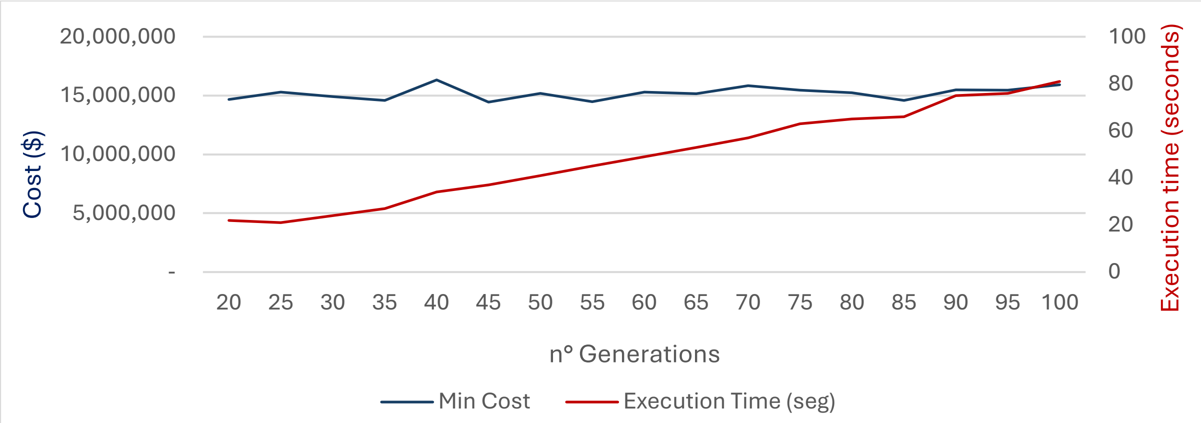
Table 12: Utilization of Production Plant A Given by the First Run of GA in Scenario 1

Utilization	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Plant A	17%	3%	3%	16%	33%	36%	56%	62%	33%	52%	59%	58%	50%	57%

Table 12 shows that Plant A has the capacity to produce all the requirements using less than the available capacity on each production line and day. At this point, a second run of the GA is performed to obtain a refined solution of the production plan. This time, we added the MOQ constraint to create a more robust production plan. The Genetic Algorithm is run again with the same parameters as in the previous execution.

We repeat the sensitive analysis for the second run of Genetic Algorithm changing the number of generations and tracking the quality of the solution and the model execution time. Testing a start at 20 generations and culminates at 100 generations. The results of this analysis are shown in Figure 10.

Figure 10: Impact of the Number of Generations on Cost Optimization and Execution Time in the Genetic Algorithm



As we can see in Figure 10, increasing the number of generations does not significantly reduce the cost of the solution. We decided to set the number of generations to 55, as the minimum cost was reached at that point within 45 seconds. It is important to note that the minimum cost obtained is higher than in the first run, due to the incorporation of additional constraints in this model. In Appendix D, we can observe how the algorithm identifies the optimal solution for the production plan by generation 20.

After this process, we also run the Genetic Algorithm to solve the distribution problem. This step takes 93 seconds.

Executing the Genetic Algorithm marks the end of the first phase of the configuration model and gives as a result a production and a distribution plan for each of the plants. As expected, there are no stockouts and inventory levels are kept above the safety stock, since in this scenario each production plant has the capacity to meet the full demand. Additionally, the production lines have sufficient capacity and are operating below full utilization. Table 13 shows a summary of the production plan for Plant A obtained from the optimization model. Table 14 shows the detail of the utilization of each production line. Table 15 shows the distribution plan from Plant A to the different distribution centers. The total execution time for this scenario was 193 seconds. The average daily cost for the production problem was $1,033,345 and for the distribution problem was $2,259,266.

Table 13: Production Plan Summary Scenario 1 (Units of Products)

Production (units)	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Production Line 2A	3,146	1,536	1,537	3,075	3,183	4,606	3,071	7,682	3,072	3,295	6,144	10,750	1,829	9,214
Production Line 3A	10,783	9,025	6,743	5,826	15,455	6,400	16,546	14,753	7,230	10,489	15,550	22,980	11,840	15,458
Production Line 6A	7,453	7,490	3,669	6,032	8,950	9,840	15,234	16,852	11,392	8,989	10,094	16,866	10,509	6,866
Total	21,382	18,051	11,949	14,933	27,588	20,846	34,851	39,287	21,694	22,773	31,788	50,596	24,178	31,538

Table 14: Line Utilization Scenario 1 (%)

Utilization (%)	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Production Line 2A	12%	4%	4%	7%	15%	11%	7%	18%	7%	23%	14%	25%	24%	21%
Production Line 3A	54%	18%	18%	11%	33%	13%	34%	31%	15%	25%	31%	49%	25%	33%
Production Line 6A	27%	16%	11%	30%	36%	44%	51%	49%	22%	27%	41%	49%	40%	31%

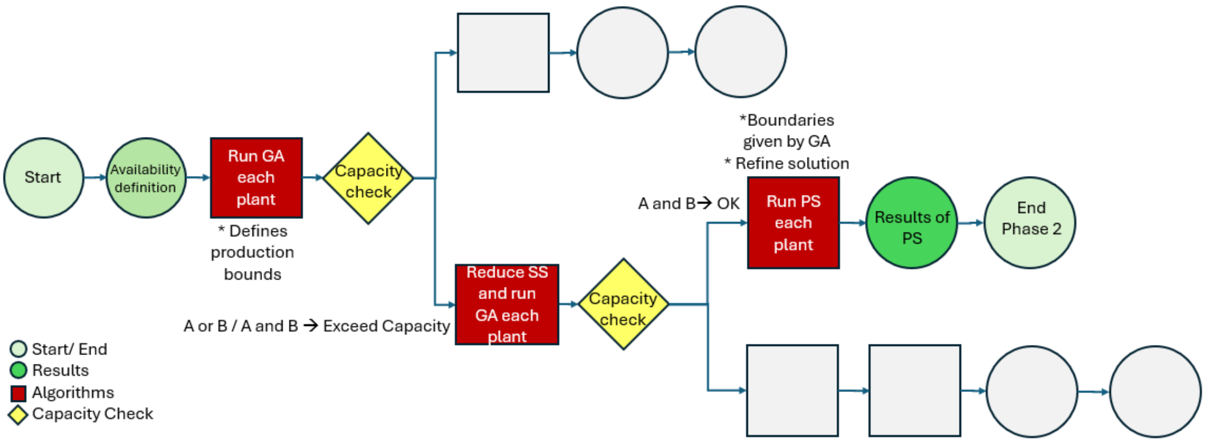
Table 15: Distribution Plan Summary Scenario 1 (Units of Products)

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
DC1A	446	2,545	865	1,366	3,177	3,877	6,719	7,104	3,719	5,502	3,410	3,750	7,063	7,481
DC2A	1,931	2,119	2,374	1,746	4,784	3,367	2,204	3,691	5,783	3,591	4,074	4,805	4,166	3,756
DC3A	241	2,640	3,403	4,243	3,235	4,402	3,072	3,473	3,272	3,420	3,768	5,700	5,402	3,819
DC4A	4,257	3,789	2,587	1,670	5,402	3,316	2,918	2,518	2,994	2,880	3,462	3,760	3,184	3,066
DC5A	57	3	18	-	-	10	6	6	8	13	-	-	23	15
DC6A	-	6	-	2	1	-	1	1	1	1	1	1	1	1
Total	6,930	11,101	9,246	9,027	16,597	14,971	14,918	16,792	15,775	15,405	14,714	18,014	19,837	18,137

4.1.2 Scenario 2: Capacity Meets Demand but Falls Below Target Inventory

The second scenario consists of having enough capacity to fulfill demand but not to replenish the safety stock. In this case, we need to check again the capacity of each plant, this time without considering the safety stock requirements. We assess this capacity using the Genetic Algorithm once again, followed by a more refined solution using Particle Swarm Optimization, as shown in colors in Figure 11.

Figure 11: Path in the Configuration of Models for Scenario 2



The main idea is that the analysis can still be performed within each production plant, without sharing resources. In this section, we carry out the analysis for Plant A. Initial parameters for this scenario are shown in Table 16.

Table 16: Parameters Scenario 2

Plant	N° SKUs	Avg Daily Demand (Units)	Initial Inventory (Units)	Inventory (Days)
A	200	64,076	508,702	7.9

Similar to Scenario 1, we follow the flow of Figure 11 by defining the availability of resources at the production plant. Then, we run the initial Genetic Algorithm to check the plant's capacity, using the same parameters defined in the previous scenario. Table 17 shows that, after running the first capacity check, it becomes clear that Plant A is not capable of meeting all the demand and safety stock requirements, as it exceeds 100% of its capacity on days 5, 6, 7, 8, 10, 11, 12, 13, and 14.

Table 17: First Capacity Check Scenario 2

Utilization	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Plant A	50%	68%	46%	99%	106%	103%	143%	138%	95%	128%	120%	143%	125%	130%

Therefore, we decided to run a second Genetic Algorithm, this time without considering the safety stock requirements. This means that only the demand will be fulfilled. The second run shows that the solution represents a compromise: meeting all the demand while covering only part of the safety stock, as shown in Table 18.

Table 18: Second Capacity Check Scenario 2

Utilization	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Plant A	65%	75%	85%	71%	86%	80%	98%	95%	85%	97%	89%	94%	96%	89%

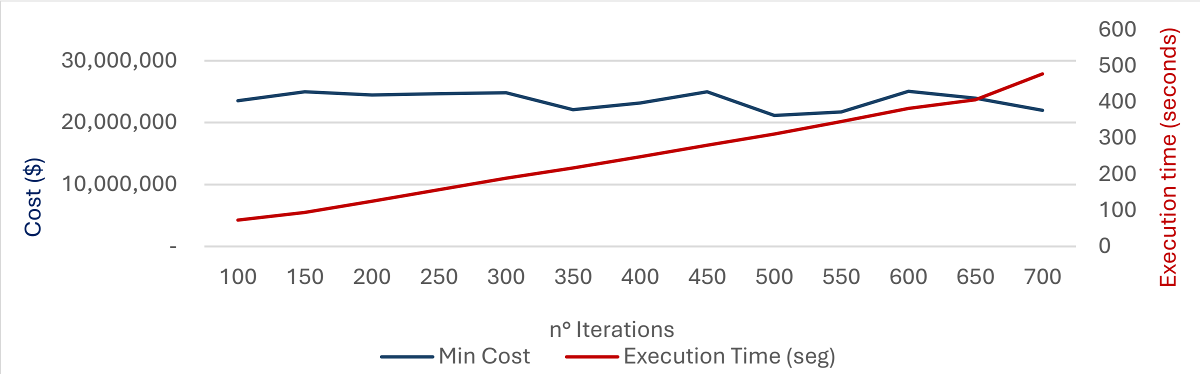
To obtain a more specific and robust solution, and to cover as much of the safety stock requirements as possible using only Production Plant A, we ran the Particle Swarm Optimization algorithm, using the results obtained from the Genetic Algorithm as boundaries. This technique allowed us to reduce the search space and find the solution more efficiently. Table 19 presents the parameters used to implement the Particle Swarm algorithm. These parameters were selected based on the recommendations of Clerc and Kennedy (2002).

Table 19: Parameters Particle Swarm Algorithm Scenario 2

Parameter	Value	Description
Population Size	20	Number of particles in the swarm
Number of Iterations	100	How many times particles update their positions
Inertia Rate	0.7	Controls momentum
Cognitive Parameter	1.5	How much a particle follows its own best solution
Social Parameter	1.5	How much a particle follows the swarm's best solution

We conducted a sensitivity analysis by varying the number of iterations in the Particle Swarm Optimization algorithm, tracking both solution quality and execution time. Figure 13 shows the change in minimum cost relative to execution time. Since we did not observe a consistent improvement in solution quality, we decided to run the model with 100 iterations. This configuration provided our solution to the production problem in 73 seconds.

Figure 12: Impact of the Number of Iterations on Optimal Solution and Execution Time in Particle Swarm Algorithm



Finally, with a clear production plan, we ran the algorithm to obtain the distribution plan for the distribution centers in 93 seconds. The outcome of this phase is a complete production and distribution plan for each plant and distribution center. Table 20 shows a summary of the production plan for Plant A obtained from the optimization model. Table 21 shows the detail of the utilization of each production line. We can see that in this scenario, since the plant cannot fully meet all the requirements, line 6A operates at full capacity on days 7, 8, and 10. Table 22 shows the distribution plan from Plant A to the different distribution centers. The total execution time for this scenario was 268 seconds. The average daily cost for the production problem was $1,682,365 and for the distribution problem was $2,985,866.

Table 20: Production Plan Summary Scenario 2 (units)

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Production Line 2A	4,681	3,072	13,934	10,862	7,680	17,006	12,581	16,896	12,398	12,837	10,825	9,436	7,973	14,117
Production Line 3A	13,925	24,142	10,990	28,768	21,952	22,003	36,916	28,256	26,560	32,041	35,598	31,767	21,152	27,703
Production Line 6A	11,214	23,185	4,971	16,671	9,263	20,803	32,639	38,328	21,168	40,779	25,960	24,229	25,743	18,428
Total	29,820	50,399	29,895	56,301	38,895	59,812	82,136	83,480	60,126	85,657	72,383	65,432	54,868	60,248

Table 21: Line Utilization Scenario 2 per Production Line (%)

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Production Line 2A	16%	7%	40%	33%	18%	47%	49%	39%	36%	67%	30%	37%	38%	53%
Production Line 3A	57%	56%	32%	58%	44%	66%	80%	60%	54%	77%	90%	69%	44%	67%
Production Line 6A	37%	58%	31%	80%	66%	86%	100%	100%	58%	100%	87%	85%	78%	69%

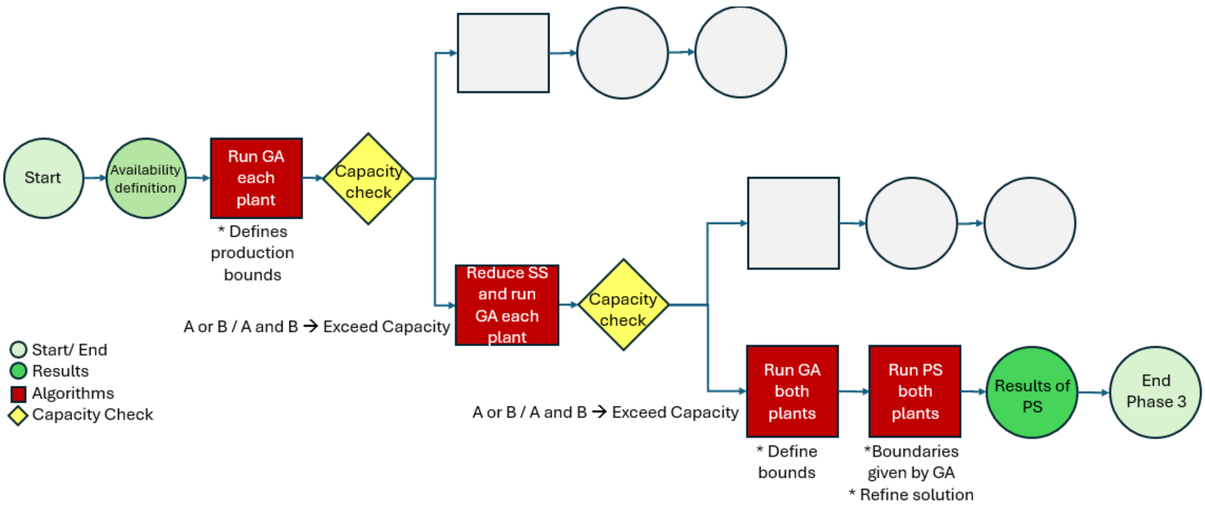
Table 22: Distribution plan summary Scenario 2 (units of products)

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
DC1A	849	5,133	1,729	2,732	6,353	7,753	13,438	14,208	7,438	11,003	6,820	7,500	14,125	14,963
DC2A	3,861	4,237	4,748	3,491	9,567	6,733	4,407	7,382	11,565	7,181	8,148	9,609	8,332	7,512
DC3A	481	5,279	6,612	8,679	6,469	8,804	6,143	6,945	6,544	6,840	7,535	11,399	10,803	7,638
DC4A	8,511	7,578	4,915	3,599	10,803	6,631	5,835	5,036	5,987	5,759	6,924	7,519	6,367	6,132
DC5A	101	17	35	-	-	20	11	11	15	25	-	-	46	29
DC6A	-	12	-	4	1	-	1	1	1	1	1	1	1	1
Total	13,803	22,256	18,039	18,505	33,193	29,941	29,835	33,583	31,550	30,809	29,428	36,028	39,674	36,275

4.1.3 Scenario 3: Inventory transfer between plants due to capacity shortfall

The third scenario consists of having insufficient capacity to fulfill demand or replenish the safety stock. In this case, resources must be shared between production plants. We assess this capacity using the Genetic Algorithm once again, followed by a more refined solution using Particle Swarm Optimization, this time applied to a more complex optimization problem that includes both production plants, as illustrated in color in Figure 13.

Figure 13: Path in the configuration model for Scenario 3



The initial parameters in this scenario consider both plants, as shown in Table 23.

Table 23: Initial parameters Scenario 3

Plant	N° SKUs	Avg Daily Demand (Units)	Initial Inventory (Units)	Inventory (Days)
A	200	73,687	508,702	6.9
B	170	36,829	259,959	7.1

Similar to Scenarios 1 and 2, we first need to define the availability of resources in each production plant. We follow the diagram of Figure 13 by running the first and second Genetic Algorithm to check the capacity of each plant. Table 24 shows the status of each production plant during the first capacity check, while Table 25 presents the status during the second capacity check, where safety stock is not considered. We can observe that in both cases, Plant A is not capable of meeting the requirements. Therefore, we designed a more complex model in which both plants can share resources.

Table 24: First capacity check Scenario 3

Utilization	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Plant A	89%	87%	62%	122%	125%	126%	173%	140%	109%	153%	146%	164%	139%	150%
Plant B	3%	10%	53%	64%	87%	85%	80%	87%	79%	98%	80%	95%	99%	78%

Table 25: Second capacity check Scenario 3

Utilization	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Plant A	89%	87%	79%	85%	102%	98%	132%	120%	101%	112%	110%	132%	112%	135%
Plant B	0%	1%	20%	48%	71%	70%	66%	67%	70%	98%	76%	95%	98%	77%

In this scenario, Plant A cannot fulfill its own demand. Therefore, we run a new optimization model with the 5 production lines of both Plant A and Plant B using the Genetic Algorithm. This step allows us to pool the demand and share resources between plants. The optimization problem becomes bigger than the previous scenarios, and the execution time increases to 90 seconds. After running the Genetic Algorithm, we run again the model using Particle Swarm for both plants to obtain a refined solution in about 90 seconds.

We use the parameters from Scenarios 1 and 2 to run both the Genetic Algorithm and Particle Swarm. In this scenario we need to balance the inventory between the two production plants since Plant B is producing to fulfill the demand in Plant A.

After running both methods, we obtain a production and a distribution plan in which plants are sharing resources. Table 26 shows the detail of the utilization of each production line in both plants. We can see that in this scenario, since Plant A cannot fully meet all the requirements, line 6A operates at full capacity on certain days. Additionally, line 1B from Plant B is also at full capacity, serving as an extra resource to support Plant A's requirements. Table 27 shows the distribution plan from Production Plant A and Production Plant B to the different distribution centers. In this case, 14,352 units are required to be transferred from Plant B to Plant A for subsequent distribution. The total execution time for this scenario was 400 seconds. Average daily cost for the production problem was $4,235,421 and for the distribution problem was $2,319,745. This cost does not include the cost of moving the inventory from Production Plant A to Production Plant B.

Table 26: Line utilization per production line Scenario 3 (%)

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Production Line 2A	12%	21%	25%	36%	29%	56%	29%	32%	56%	56%	20%	60%	38%	38%
Production Line 3A	50%	38%	66%	59%	59%	50%	67%	80%	74%	80%	74%	67%	57%	53%
Production Line 6A	24%	35%	77%	89%	69%	71%	100%	100%	81%	100%	92%	87%	76%	76%
Production Line 1B	5%	18%	87%	93%	93%	100%	99%	100%	93%	100%	100%	100%	83%	90%
Production Line 2B	5%	18%	46%	41%	37%	62%	44%	82%	55%	49%	58%	49%	39%	48%

Table 27: Distribution plan summary Scenario 3 (units of products)

		Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Plant A	DC1A	1,614	5,275	2,007	3,606	7,462	11,587	16,452	16,647	8,877	12,708	8,053	8,795	16,414	17,211
	DC2A	2,433	4,292	3,909	3,486	10,193	10,831	7,932	9,237	16,494	9,475	10,420	11,516	9,201	9,389
	DC3A	687	5,962	6,732	7,611	5,395	10,100	7,151	10,006	10,772	10,434	10,952	13,692	12,894	9,947
	DC4A	6,887	8,604	4,033	4,393	10,209	9,807	7,884	8,062	10,142	7,887	7,869	9,864	7,500	7,259
	DC5A	64	6	36	-	-	18	20	20	17	29	-	-	35	40
	DC6A	-	15	-	5	1	-	1	1	1	1	1	1	1	1
Plant B	DC1B	10,497	7,403	13,350	7,687	9,265	10,037	6,955	29,811	22,576	10,805	9,244	11,070	10,471	11,160
	DC2B	3,010	7,488	9,727	2,599	5,561	6,973	6,862	6,199	8,347	6,063	6,393	6,722	8,492	6,641
	DC3B	-	32	-	2	1	44	-	-	13	-	-	98	6	-
	DC4B	3	55	-	150	195	278	319	356	261	25	246	319	457	222
	DC5B	1,860	2,078	2,911	5,615	4,152	5,256	4,372	4,121	7,703	6,000	4,683	5,467	7,418	4,278

4.2 Summary and Model Comparison

In this section, we compare our configuration of models with the company's current approach to solving the production and distribution optimization problem in terms of execution time. To test the quality of the model, we focus on comparing our Scenario 2 with the company’s solution, as this is the only available data we have.

4.2.1 Execution Time Summary

Table 28 represents a summary of the execution time in each step in each scenario analyzed. We can observe that in the most complex scenario (Scenario 3) the execution time is less than 7 minutes (400 seconds).

Table 28: Execution time summary

Step	Description	Execution Time (seconds)		
		Scenario 1	Scenario 2	Scenario 3
1	Availability Definition			
2	Genetic Algorithm in each plant for first capacity check and set boundaries	50	50	50
3	Genetic Algorithm in each plant for a refine solution	50	NA	NA
4	Genetic Algorithm for distribution problem	93	NA	NA
5	Genetic Algorithm in each plant for second capacity check	NA	50	50
6	Particle Swarm Algorithm in each plant for a refine solution	NA	73	NA
7	Particle Swarm Algorithm in each plant for distribution problem	NA	95	NA
8	Genetic Algorithm in both plants for set boundaries	NA	NA	90
9	Particle Swarm Algorithm in both plant for a refine solution	NA	NA	90
10	Particle Swarm Algorithm in both plant for distribution problem	NA	NA	120
	Total	193	268	400

4.2.2 Model Comparison

We compare our results in Scenario 2 with the company’s solution in terms of both solution quality and execution time. Tables 29 and 30 show the line utilization in our solution and the company's solution, respectively.

Table 29: Line utilization per production line Scenario 2 (%)

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Production Line 2A	16%	7%	40%	33%	18%	47%	49%	39%	36%	67%	30%	37%	38%	53%
Production Line 3A	57%	56%	32%	58%	44%	66%	80%	60%	54%	77%	90%	69%	44%	67%
Production Line 6A	37%	58%	31%	80%	66%	86%	100%	100%	58%	100%	87%	85%	78%	69%

Table 30: Line utilization per production line Company´s solution (%)

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Production Line 2A	4%	18%	4%	18%	18%	18%	30%	9%	34%	42%	26%	8%	22%	46%
Production Line 3A	42%	21%	74%	62%	74%	91%	72%	65%	81%	88%	71%	74%	28%	100%
Production Line 6A	65%	90%	80%	81%	65%	75%	83%	73%	75%	78%	71%	68%	82%	31%

We observe that our solution balances the capacity across production lines and leverage the available capacity more efficiently. In terms of units produced, our solution achieves a 13% increase compared to the company's solution, while also reducing the penalty cost in the production problem by 2%, considering overstock penalties, penalties for inventory below the minimum, penalties for inventory below the target, and stockout penalties. This improvement is mainly because the optimization model penalizes stockouts 1,500 times more heavily than overstock situations. Additionally, in terms of distribution costs, our solution is 13% lower than the company's, again driven by the strong penalty associated with stockouts.

Another way to measure the quality of our model is by comparing the resulting inventory positions. Tables 31 and 32 show the number of SKUs in Plant A projected to be in stockout, below the minimum inventory level, below the target inventory level, and above the target inventory level in both our solution and the company's solution.

Table 31: Inventory position Scenario 2 (SKUs)

SCENARIO 2	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Over Target	200	200	200	200	200	197	192	189	190	185	186	185	183	186
Between Target and Minimum	-	-	-	1	-	1	4	5	5	2	5	3	3	1
Between Minimum and 0	1	-	-	-	-	-	1	4	-	2	3	3	1	3
Below 0	-	-	-	-	-	-	-	-	1	-	-	-	-	-
Total	200	200	200	200	200	200	200	200	200	200	200	200	200	200

Table 32: Inventory position Company´s solution (SKUs)

COMPANY	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
Over Target	186	189	190	189	191	186	186	183	185	182	182	180	184	182
Between Target and Minimum	5	6	5	7	4	7	8	9	5	7	6	7	5	4
Between Minimum and 0	5	2	2	1	2	3	2	4	6	7	8	9	7	9
Below 0	4	3	3	3	3	4	4	4	4	4	4	4	4	5
Total	200	200	200	200	200	200	200	200	200	200	200	200	200	200

As shown in Table 31 and Table 32, our solution aims to reduce the number of SKUs that fall below the target level, below the minimum, and into stockout, compared to the company’s solution. Another important consideration is that our model in Scenario 2 takes only 268 seconds to run, compared to approximately half an hour (1800 seconds) for the company's solution.

For an excerpt of the production and distribution plan obtained with the proposed model, refer to Appendix E and F.

4.3 Model Limitations

Since our model is designed to be fast and provide a feasible solution in a short period, it also comes with some limitations. As mentioned in Section 2.3, using the Genetic Algorithm and the Particle Swarm does not always guarantee finding the global optimum, but they produce a high-quality solution if parametrized correctly. When capacity is not a constraint, a small discrepancy from the optimal solution may be acceptable; however, when capacity is limited, finding the optimal solution becomes crucial. Therefore, the ideal scenario would be to be able to use both models according to the situation. On the one hand, using the MIP model for regular planning when time is not a constraint would grant optimality. On the other hand, the combination of both algorithms can deliver valuable insights for the planning team, especially when the situation requires a quick response.

Another important limitation is that the performance of both algorithms heavily depends on proper parameter tuning, such as population size, number of generations, mutation and crossover rates, and number of iterations, among others. In our case, we evaluated the behavior of the optimal solution under specific scenarios, but it is important to note that more detailed parameter tuning would be required in the case of a broader variety of scenarios.

It is also important to mention that due to random initialization, different runs may yield different solutions, affecting repeatability. Because of this, it is sometimes difficult to interpret the solutions and justify decision-making, as it is not always clear how the algorithms operate or reach those results.

As mentioned in Section 1.4, the sponsor company is currently working with cost figures that represent penalties to drive the desired behavior into the model. Therefore, the models do not calculate the real cost of manufacturing and delivering the product, as is shown in Figure 3: Framework Project. We believe that there could be significant benefits from using estimated values, such as obtaining better insights into the business and making decisions to reduce the bottom line. In addition, currently the sponsor company is not considering the production (manufacturing) and transport costs. In a similar way, adding transportation costs could help the company understand a trade-off between stocking more inventory and transferring inventory from another plant.

Finally, it is also important to mention that there is data preparation required to execute the program that we built. Currently, OMP connects directly to SAP and extracts the information it needs to run the optimization automatically. In our case, we had to extract the information from the system and upload that information into Google Colab to run the program in Python. In this document, we are comparing the execution time of the MILP-based program against our configuration of metaheuristic models, but we are not including the time to obtain and prepare the information. Since our model cannot be directly connected to the company's ERP, we would recommend automating the process to obtain the necessary data to run the program.

5. CONCLUSION

To conclude our capstone, this chapter shares key management takeaways based on the model's results, along with some suggestions for future improvements.

5.1 Insights and Recommendations

The sponsor company is currently using Mixed-Integer Programming (MIP) based software to optimize its end-to-end supply chain planning. The main purpose of the company is to increase the speed at which it responds to changes in demand and external conditions. To achieve this, the sponsor company wants to execute their optimization planning program more quickly, and as a result, being able to update demand and production conditions as needed.

To accomplish such an aim, we propose two key approaches, based on our research and analysis. The first relates to the size of the problem; it is possible to break down the large optimization problem into smaller, more manageable sub-problems while still respecting all constraints. Specifically, in this project, we broke the problem based on three criteria: location (Plant A and Plant B), process type (production and distribution), and equipment used (Product Type Blue and Product Type Red). However, it is essential to conduct a rigorous evaluation of the supply chain before applying this strategy, as it requires identifying all interdependencies between variables and business-specific features.

The second approach is the potential use of alternative optimization methods that can deliver high-quality solutions more quickly than traditional MIP. In our study, we evaluated a hybrid approach that combines two metaheuristic algorithms, Genetic Algorithm and Particle Swarm Optimization, and leveraged their strengths to design a configuration model capable of delivering fast solutions that are close to optimal. Additionally, we found that providing an estimated baseline solution significantly reduces the execution time of the metaheuristic methods.

Regarding the overall process agility, we recommend limiting the planning horizon depending on its purpose and adjusting the frequency accordingly. There are traditionally three-time horizons for supply chain planning (long-, medium-, and short-term). While running a planning process twice per day is important to respond to changes in demand on an operational level (short-term), it does not add value on the tactical level (medium-term). We consider that the short-term horizon should be analyzed daily and independently, especially when there are disruptions in supply or changes in demand. However, the medium-term horizon should only be run once or twice per month to make tactical decisions.

The final insight is that a well-tuned metaheuristic model configuration can outperform a MIP model when the latter is prematurely terminated. Since the sponsor company enforces a 1.5-hour time limit on the planning program execution, the program returns the best solution that has been calculated so far. Therefore, even when working with a MIP model, optimality is not guaranteed. In our experiments, the proposed solution reduced production costs by 2% and distribution costs by 13%. These improvements are driven by lowering the number of SKUs under the inventory target, under the minimum target, and inventory stockouts. While the overall savings are modest, they were achieved in 85% less time than the current MIP process.

5.2 Broader Applicability of the Developed Optimization Models

In global consumer goods enterprises, uncertainty in demand, supplier performance, and external factors can lead to poor inventory planning, resulting in overstock or stockouts. To mitigate these risks, companies must react quickly and responsively to any environmental changes that may disrupt production or distribution plans. However, creating a new production or distribution plan can be difficult and time-consuming, especially in companies managing a large number of SKUs.

Through this study, we present an alternative, near-optimal approach to solving optimization problems. While traditional MIP-based solutions are effective in many cases, other methodologies can offer improvements in both speed and solution quality. Several of these methods can be integrated into existing systems to deliver results that are both time-efficient and high in quality.

The methodology presented in this study can be applied not only to production and distribution planning, but also to a wide range of optimization problems within the supply chain, such as storage, transportation, and procurement, across various industries. Adaptation is possible by modifying specific parameters.

The conclusions and insights found in this study are valuable not only for our sponsor company but also for other industries facing similar challenges. These results are encouraging for organizations to explore alternative approaches to solve complex optimization problems.

REFERENCES

Accenture. (2020). Building a consumer-centric supply chain in consumer goods. https://www.accenture.com/content/dam/accenture/final/a-com-migration/pdf/pdf-144/accenture-building-a-consumer-centric-supply-chain-in-consumer-goods.pdf

Adiche, Chahrazad, and Méziane Aïder. (2010) "A Hybrid Method for Solving the Multi-objective Assignment Problem." (2010). 154-157

Banker, S. (2022). Supply Chain Planning For Dummies. *Forbes.* https://www.forbes.com/sites/stevebanker/2022/02/04/supply-chain-planning-for-dummies/

Boyabatli, O., & Sabuncuoglu, I. (2004). Parameter Selection in Genetic Algorithms. *Journal of Systems Cybernetics and Informatics*. 4, (2), 78-83.

Burnette, M., Dittmann, P. (2018). End-to-End Supply Chain Collaboration. 15 - 21 https://supplychainmanagement.utk.edu/wp-content/uploads/2022/03/2018-05-End-to-End-Supply-Chain-Collaboration-Best-Practices.pdf

Chopra, S., & Meindl, P. (2015). *Supply chain management: Strategy, planning, and operation* (6th ed.). Pearson.

Christopher, M., & Towill, D. R. (2000). Supply chain migration from lean and functional to agile and customized. *Supply Chain Management: An International Journal*, *5*(4), 206–213.

Clerc, M., & Kennedy, J. (2002). The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, *6*(1), 58–73. https://doi.org/10.1109/4235.985692

Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. John Wiley & Sons.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *26*(1), 29–41.

Dzalbs, I., & Kalganova, T. (2020). Accelerating supply chains with ant colony optimization across a range of hardware solutions. In *2020 International Conference on UK-China Emerging Technologies (UCET)* (pp. 1–4). IEEE. https://bura.brunel.ac.uk/handle/2438/20185

Esmaeilikia, M., Fahimnia, B., Sarkis, J., Govindan, K., Kumar, A., & Mo, J. (2016). *Tactical supply chain planning models with inherent flexibility: Definition and review. Annals of Operations Research, 244(2), 407–427. https://doi.org/10.1007/s10479-014-1544-3*

Ghandour, J., Lange, T., & Turco, A. (2021). *Consumer-goods companies must transform their planning end-to-end*. McKinsey & Company. https://www.mckinsey.com/industries/consumer-packaged-goods/our-insights/consumer-goods-companies-must-transform-their-planning-end-to-end#/

Ho, K., Shimane, Y., & Isaji, M. (2024). *Generalizing Space Logistics Network Optimization with Integrated Machine Learning and Mathematical Programming*

Innocente, M., & Sienz, J. (2010). Coefficients' settings in particle swarm optimization: Insight and guidelines. In E. Dvorkin, M. Goldschmit, & M. Storti (Eds.), Mecánica Computacional:

Computational intelligence techniques for optimization and data modeling. Asociación Argentina de Mecánica Computacional. https://arxiv.org/pdf/2101.11944

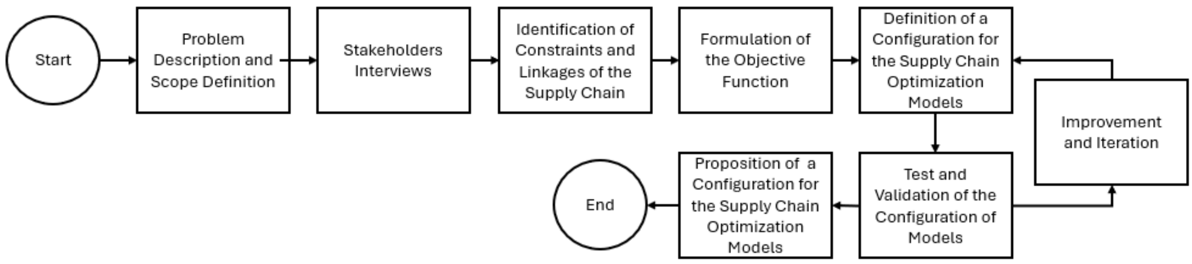
Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2007). Designing and managing the supply chain: Concepts, strategies and case studies (3rd ed.). McGraw-Hill.

Suri, R. (1998). *Quick response manufacturing: A companywide approach to reducing lead times*. Productivity Press.

Triantafyllou, N., & Papathanasiou, M. (2024). *Deep learning enhanced mixed integer optimization: Learning to reduce model dimensionality*.

APPENDICES

Appendix A: Process to define the Configuration of Models



Appendix B: Other metaheuristic techniques

Tabu Search

Tabu Search is a method that can be applied for the optimization of complex problems with multiple objectives. Tabu Search is a metaheuristic optimization method that begins with local search by incorporating memory structures to explore the solution space and avoid returning to previously visited solutions (Glover, 1986).

The methodology starts with a feasible solution that is often generated randomly or by using a heuristic method. It is necessary to define a Tabu list, which represents the recently visited solutions or forbidden moves. It is also necessary to define the Tabu list size and one stopping criterion in the model to conclude the optimization. After these definitions, it is essential to determine the neighborhood, where all the solutions can be reached, and evaluate all possible solutions in the neighborhood. The algorithm will select the best move from the neighborhood based on the objective function, and every new movement must be checked against the forbidden list. The current solution will be replaced with the new selected movement that satisfies the objective function. The search ends when the stopping criterion is met (Glover, 1986).

Simulated Annealing

Simulated Annealing is an analogy of the annealing process, which refers to a metal freezing into a minimum energy structure. The advantage of this method over other optimization models is its effectiveness in looking for the global minimum of the objective function. This quality is achieved by accepting changes that decrease the objective function, and changes that increase them with a probability:

$$p = exp\frac{-\delta f}{T} \qquad \text{(Eq 1)}$$

where is the change in the objective function and is the control parameter (stands for temperature, referencing the annealing process).

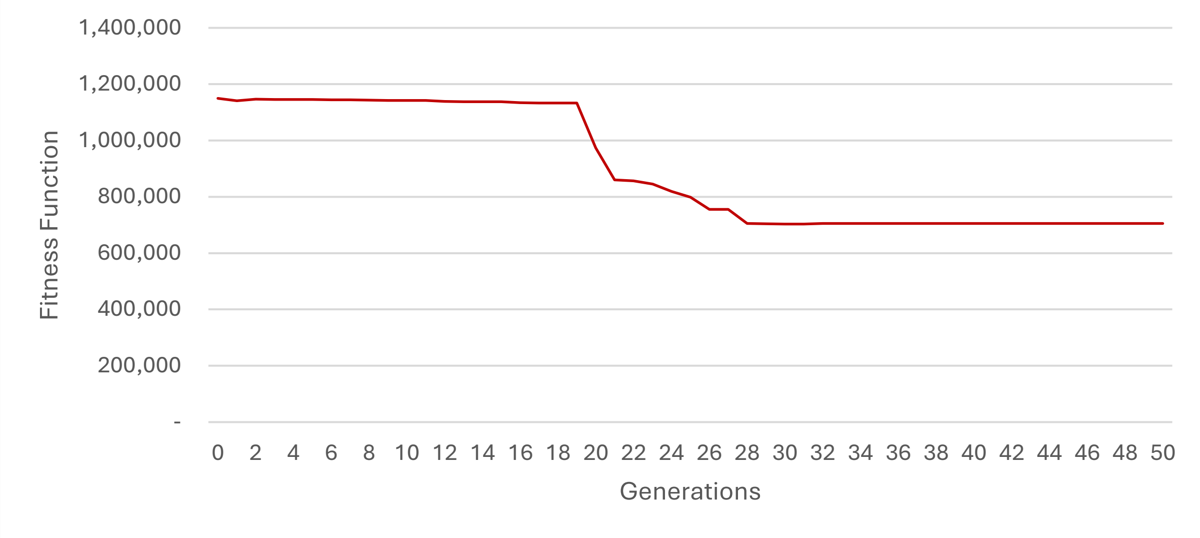
The process consists in creating an initial solution and estimating its initial *T*. Then, we need to create a new solution with a generator of random changes that allows all possible solutions to be reached. The new solution should be evaluated against the current one, and if it is accepted, it becomes the current solution. The *T* parameter is adjusted according to an annealing schedule, and the process is repeated a number of times defined by the required amount of *T* values to be used, or a total number of solutions to be generated. (Parks & Sepulchre, 2020).

Ant Colony Optimization

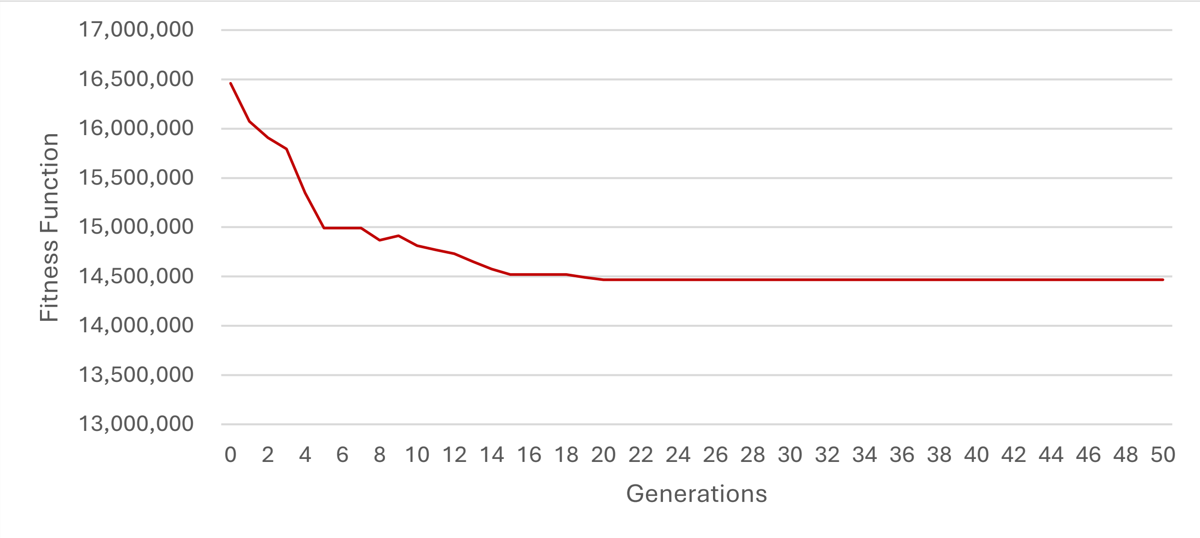
The ant colony algorithm is an optimization method based on the behavior of real ants within colonies, with the objective of solving a wide range of optimization problems. The algorithm mimics the search patterns between the colony and food. The main idea is that ants deposit pheromones to delimit different paths, which serve as guidance to other ants to find optimized paths or solutions (Dorigo and Stützle, 2004). The communication over iterations via pheromone left from previous ants allows better solutions to be found.

The method begins with the definition of a certain number of pheromones as an initial parameter in the paths (solutions). The algorithm allows other ants to use these pheromones as a signal and follow the paths with higher concentrations. The right path (the decision rule) is chosen considering the pheromone intensity (path with higher concentrations are more attractive) and heuristic information such as local factors (e.g., distance between nodes). More desirable solutions or paths attract more ants and lead to higher probabilities to select a specific solution. Once all ants have completed their path decisions, the pheromones are updated to reflect the quality of the solutions. Over time, the evaporation of pheromones occurs. This phenomenon helps to avoid convergence of suboptimal solutions and ensures that ants continue exploring new paths. The process of solution creation and information update is repeated for a predefined number of iterations or until certain criteria are met (Doringo, 1996).

Appendix C: Fitness Function throughout Generations Genetic Algorithm



Appendix D: Fitness Function throughout Generations Genetic Algorithm



Appendix E: Production Plan proposal Model

SKU	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	110	0	0	0	0	110	0	0	0	110	0
22	0	0	0	0	0	0	110	0	110	0	110	0	110	110
23	1729	0	1536	0	0	0	1536	0	0	1536	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	73	0	0	73
26	0	0	0	0	0	0	0	0	0	1536	0	0	0	0
27	0	0	0	0	0	0	0	0	0	73	0	73	0	0
28	1536	0	1536	0	0	0	0	0	1536	0	0	0	1536	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	1536	0	0	1536	0	0

Appendix F: Distribution Plan Proposal Model

SKU	Location	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
1	DC1A	10	247	56	56	0	79	56	28	0	0	0	0	0	0
2	DC2A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	DC3A	15	196	28	28	84	55	84	28	0	0	0	0	0	0
4	DC2A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	DC1A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	DC2A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	DC1A	0	0	0	0	7	0	0	0	20	0	0	0	0	0
8	DC2A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	DC1A	13	4	12	6	8	19	95	25	64	23	28	19	29	41
10	DC3A	0	4	1	0	128	9	39	27	51	13	31	67	46	52
11	DC4A	0	20	0	8	44	5	24	19	17	12	26	44	8	8
12	DC5A	0	0	27	1	129	11	160	23	39	46	39	53	50	65
13	DC2A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	DC1A	0	0	0	0	0	0	0	0	219	192	0	608	0	0
15	DC2A	0	0	0	0	0	0	0	0	227	0	0	146	0	0
16	DC3A	0	0	0	0	0	0	0	0	83	0	0	0	0	0
17	DC4A	0	4	32	1	4	17	13	32	30	19	24	32	17	14