# MATHEMATICAL MODEL AND GENETIC ALGORITHM FOR THE INTEGRATED PROBLEM OF PRODUCTION NETWORK DESIGN AND INVENTORY POSITIONING

**BY**

**NGUYEN TRONG TRI DUC**

**A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY (ENGINEERING AND TECHNOLOGY)**

**SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY**

**THAMMASAT UNIVERSITY**

**ACADEMIC YEAR 2023**

THAMMASAT UNIVERSITY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

DISSERTATION

BY

NGUYEN TRONG TRI DUC

ENTITLED

MATHEMATICAL MODEL AND GENETIC ALGORITHM
FOR THE INTEGRATED PROBLEM OF
PRODUCTION NETWORK DESIGN AND INVENTORY POSITIONING

was approved as partial fulfillment of the requirements for
the degree of Doctor of Philosophy (Engineering and Technology)

on January 24, 2024

Chairperson ______________________________
(Associate Professor Parthana Parthanadee, Ph.D.)

Member and Advisor ______________________________
(Associate Professor Jirachai Buddhakulsomsiri, Ph.D.)

Member ______________________________
(Associate Professor Pisal Yenradee, D.Eng.)

Member ______________________________
(Associate Professor Warut Pannakkong, Ph.D.)

Member ______________________________
(Assistant Professor Pham Duc Tai, Ph.D.)

Director ______________________________
(Professor Pruettha Nanakorn, D.Eng.)

| | |
|---|---|
| Dissertation Title | MATHEMATICAL MODEL AND GENETIC ALGORITHM FOR THE INTEGRATED PROBLEM OF PRODUCTION NETWORK DESIGN AND INVENTORY POSITIONING |
| Author | Nguyen Trong Tri Duc |
| Degree | Doctor of Philosophy (Engineering and Technology) |
| Faculty/University | Sirindhorn International Institute of Technology/ Thammasat University |
| Dissertation Advisor | Associate Professor Jirachai Buddhakulsomsiri, Ph.D. |
| Academic Years | 2023 |

# ABSTRACT

This research presents integrated approaches to address the production network design and inventory positioning problem within multi-product and multi-period contexts. Our objective is to concurrently determine the network structure, safety stock amounts, and their respective locations while accounting for normal demand distribution. The study aims to minimize the overall production network cost. This approach combines traditional network design formulation with inventory positioning concepts utilizing the guaranteed service approach. We apply the proposed model to a numerical study involving office furniture manufacturing, considering the bill of materials. Our approach's effectiveness is demonstrated through comparison with a sequential strategy, where network structure and safety stock decisions are made sequentially. Results show the integrated approach outperforms the sequential one, with cost savings ranging from 1.7% to 3.7%. Furthermore, the integrated approach yields higher optimal cycle service levels. To assess model performance, a sensitivity analysis on critical parameters is conducted, revealing insights into the influence of committed service time and demand variation coefficient on solutions. Notably, longer committed service times result in reduced safety stock costs and higher optimal service levels. This leads to allocating more safety stock to upstream components compared to finished goods. Additionally, an innovative genetic algorithm is proposed for dealing with the integrated problem to tackle the problem's nonlinearity and complexity with different sourcing strategies, where the concept of rank-based decoding within the genetic algorithm is introduced. This decoding technique transforms the network design component into a simplified minimum-cost flow problem. By leveraging mixed-integer linear programming models, we solve the minimum-cost flow and inventory positioning problems. We validate our genetic algorithm solution by comparing it with a commercial solver's solutions, which are optimal for medium-sized instances, or the best found for large ones. Results demonstrate the genetic algorithm's capability to attain optimal solutions for medium-sized problems and outperform the best-found solutions for large-sized problems.

**Keywords**: Genetic algorithm, Guaranteed-service approach, Mixed-integer linear programming, Inventory positioning, Integrated approach, Network design, Rank-based decoding, Sourcing strategies

# ACKNOWLEDGEMENTS

Time has swiftly passed, marking the end of my enriching five-year expedition at SIIT, Thammasat University. My voyage as a Ph.D. candidate has been a truly gratifying experience. I am deeply grateful for the many individuals who have supported me wholeheartedly on this journey. I'd like to convey my sincerest thankfulness to those who have infused my journey with excitement and joy.

Foremost and with the utmost gratitude, I extend my thanks to my esteemed Ph.D. advisor, Assoc. Prof. Dr. Jirachai Buddhakulsomsiri. I am indebted for the unwavering support and creative latitude that Ajarn Jirachai granted me, enabling the pursuit of my research passions. His guidance on both research and the path ahead has indelibly influenced me.

Gratitude flows towards my dissertation committee members: Assoc. Prof. Dr. Parthana Parthanadee, Assoc. Prof. Dr. Pisal Yenradee, Assoc. Prof. Dr. Warut Pannakkong, and Asst. Prof. Dr. Pham Duc Tai. With their invaluable assistance, this dissertation found its fruition.

My Ph.D. journey would have been incomplete without the mentorship of many others who illuminated my path over the years. Asst. Prof. Dr. Pham Duc Tai deserves my appreciation for his unyielding support and research insights. His guidance has been a cornerstone of my growth and development in this pursuit.

A luminous moment in my Ph.D. narrative was the enriching month spent at Hokkaido University during the PARE program. My gratitude encompasses all those who made this journey possible, particularly my advisor, Ajarn Jirachai, and the SIIT staff. This precious opportunity allowed me to rejuvenate after the research's vicissitudes. I wish to convey my sincere gratitude to the countless colleagues and friends I had the privilege to meet at Hokkaido University. Their shared joys and enriching experiences have significantly enhanced my internships, and I am genuinely thankful for their presence in my journey.

To the many comrades whose names elude this note's brevity, I express my thanks.

Lastly, my eternal gratitude extends to my parents, sister, brother-in-law, niece, nephew, and relatives for their unwavering support, enabling me to pursue my life's passion.

Nguyen Trong Tri Duc

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS/ABBREVIATIONS

| **Symbols/Abbreviations** | **Terms** |
|---|---|
| ANOVA | Analysis of Variance |
| BOM | Bill of Materials |
| CSL | Cycle Service Level |
| CV | Coefficient of Variation |
| FLP | Facility Location Problem |
| GA | Genetic Algorithm |
| GSA | Guaranteed-Service Approach |
| IP | Inventory Positioning |
| MCF | Minimum-Cost Flow |
| MILP | Mixed-Integer Linear Programming |
| ND | Network Design |
| NDP | Network Design Problem |
| RT | Replenishment Time |
| SCND | Supply Chain Network Design |
| SC | Supply Chain |
| SD | Standard Deviation |
| SL | Service Level |
| SS | Safety Stock |

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction

Supply chain (SC) management is a vital field encompassing various decisions that impact operational efficiency, customer satisfaction, and financial performance. Among these decisions, supply chain network design (SCND) and inventory positioning (IP) stand out as crucial elements that influence the performance of a supply chain. Both these areas involve complex considerations, requiring innovative approaches for effective decision-making.

Network design (ND) is a critical problem of SC management involving strategic planning that determines locations, capacities, and material flows within the network. These decisions are characterized by their long-term impact and are expensive to change, including supplier selection, facility location and capacities decisions, and product flow determination to optimize the overall cost while maintaining service levels (SLs). When designing a production network, essential factors, such as market demand, production capacity, sourcing strategies, lead times, and distribution channels are considered. On the contrary, tactical decisions, spanning a medium-term timeframe, concern aspects associated with capacity allocation and the placement of safety stock (SS). To address these effectively, organizations often establish two distinct SC functions: one for strategic planning and the other for tactical planning.

Strategic and tactical planning within SC management can be handled in various ways. One method is utilizing an integrated model, also known as a monolithic model, where both strategic and tactical decisions are addressed simultaneously (Weintraub & Navon, 1976). Alternatively, a hierarchical model can be adopted, involving sequential consideration of the two functions. This can include a single iteration, as outlined by Smith (1978), or multiple iterations, as explored by Weintraub & Cholaky (1991). Hierarchical models are often favored for their simplified solving process. Nonetheless, it's vital for these functions to interact closely, as strategic decisions establish constraints for tactical planning, and tactical decisions determine the resources needed for higher-level strategic choices (Hax, 1973).

When establishing an SC, the placement of SS holds significant impacts on a company's financial and operational aspects. Simchi-Levi et al. (2008) emphasize that IP involves determining the optimal locations for SSs within a network to minimize the overall cost associated with SSs while ensuring a satisfactory level of customer service. In essence, SSs act as a buffer against demand uncertainty and lead times, protecting the SC from potential stockouts. To achieve an optimal solution, it is crucial to consider various factors influencing SS placement thoughtfully. These include desired cycle service levels (CSL), the SC network structure, net replenishment times (RT), as well as operational considerations and constraints. To this end, a strategic model that effectively integrates appropriate SS positioning becomes essential for management. However, despite this significance, SCND and IP problems are often tackled in isolation due to their apparent simplicity. However, it is essential to recognize that decisions concerning facility openings and SS placements are interconnected (Puga et al., 2019). Integrating these choices has the potential to yield a more effective SC network. Nevertheless, this integration has received limited attention within the operations management and SC planning research communities, possibly due to the intricate nature of the challenge.

While various models exist for addressing SCND and IP separately, this study focuses on an integrated approach that unifies these decisions. Integrated models simultaneously consider strategic and tactical decisions, recognizing their inherent interdependencies. This approach is aimed at achieving greater efficiency and effectiveness in the operation and performance of SC.

As businesses increasingly recognize the vital role of sourcing in their long-term operations, various sourcing strategies come into play. Two primary strategies employed are multiple and single sourcing. Single sourcing refers to procuring all required items from a single supplier (Treleven & Schweikhart, 1988). This approach offers simplicity, potential cost savings through quantity discounts, and the cultivation of stronger supplier relationships. Conversely, multiple sourcing involves acquiring items from multiple suppliers (Seshadri et al., 1991). In contrast to single sourcing, multiple sourcing enhances resilience by mitigating the impact of supply disruptions (Treleven & Schweikhart, 1988). Both multiple and single-sourcing strategies are considered in our integrated problem study to reflect real-world practices.

Addressing such integrated problems using conventional commercial solvers often proves ineffective and time-consuming, especially when dealing with large-scale instances, often resulting in inefficiencies and suboptimal solutions. Consequently, metaheuristics emerge as a viable solution approach. Metaheuristics are potent problem-solving methods well-suited for addressing complex and large-scale optimization challenges. Their adaptability and effectiveness within the SC domain have been proven in various research (Gutierrez et al., 2018; Nezamoddini et al., 2020; Robles et al., 2020; Darmawan et al., 2021; Gholizadeh et al., 2022), enabling researchers to make informed decisions and optimize systems with enhanced efficiency and performance.

This paper delves into the complexities of the integrated SCND and IP problem, proposing novel approaches to enhance decision-making. Through this research, a deeper understanding of the relationship between ND and IP is indicated, offering valuable insights for enhancing SC operations in a rapidly evolving business landscape.

## 1.2 Objectives

Our study introduces an innovative approach to address the complex challenge of integrating the NDP with the IP problem. The key objective is to develop an integrated model that jointly determines crucial aspects such as facility opening and capacity selection decisions, product flows, demand allocations, and SS placements. This integrated model considers multi-echelon, multi-product scenarios and incorporates the bill of materials (BOM), making it highly relevant to real-world SC.

To model the IP problem, we employ the guaranteed-service approach (GSA), assuming deterministic service times between orders placed at the upstream supplier and their releases to the downstream customer. The GSA distinguishes between two types of demand uncertainty: bounded and unbounded. SS handles bounded uncertainty, while measures such as expedited shipments and outsourcing address unbounded uncertainty, ensuring consistent fulfillment time and service.

Our primary contributions are as follows.

- Development of a comprehensive mathematical programming model that integrates ND and IP while considering normal demand assumptions.

- Linearization of the non-linear IP problem, involving SS calculations through the determination of RTs. Moreover, due to the interdependence between network structure and SS placement, determining SS locations becomes challenging if the network structure is not known in advance. Our approach addresses this issue by reformulating nonlinearities into a mixed-integer linear programming model (MILP).
- Extension of a simple GSA-based model proposed to accommodate multiple products with BOM characteristics, allowing flexible SS placement across the network.
- Optimization of the CSL to minimize total stockout costs as part of the overall SS cost, leading to lower SS cost.
- A case study involving a chair manufacturing network is presented to validate the practical application of the developed model. Comparative analysis against separate ND and IP models indicates the superior performance of our proposed integrated model regarding cost savings.
- Introduction of a Genetic Algorithm (GA) for the integrated problem, considering both multi-sourcing and single-sourcing strategies, is necessitated by two main reasons. Firstly, the complexities in addressing large-scale problems make it challenging when solved by commercial solvers. Secondly, the intricate nature of the multi-sourcing strategy makes it unfeasible to develop an integrated MILP model for this particular sourcing strategy.
- Proposing a novel rank-based decoding procedure for the GA, enabling the use of decoded results for the NDP, resulting in a simplification of the ND model into a minimum-cost flow problem.
- Through extensive numerical experiments, our results highlight the GA's ability to obtain optimal solutions for medium-scale problems. Additionally, a fine-tuned GA demonstrates its capability to come up with effective solutions for large-scale problems under different sourcing strategies.

## 1.3 Dissertation organization

The subsequent sections of the dissertation are organized as follows. Chapter 2 provides an overview of relevant literature related to our study. In addition, Chapter 3 presents the development of the integrated model of ND and IP problems. Chapter 4 shows the development of our GA for the problem. A numerical example and results comparison are displayed in Chapter 5. Chapter 6 is to conclude our research and offers recommendations for potential studies in the future.

# CHAPTER 2
# REVIEW OF LITERATURE

## 2.1 Inventory positioning problem

This research involves determining optimal SS locations and quantities within an SC – often referred to as the IP or SS placement problem. The problem is commonly addressed through either the stochastic service approach (SSA) or the guaranteed service approach (GSA). SSA and GSA were introduced by Clark & Scarf (1960) and Simpson (1958), respectively, with a fundamental distinction in how they manage demand uncertainty.

In SSA, the sole means of dealing with uncertainty is SS (Clark & Scarf, 1960). In the event of a shortage, it is backordered and met when inventory becomes available. This lack of alternative action during a shortage leads to random demand fulfillment timing and results in a stochastic SL. On the other hand, GSA categorizes demand uncertainty into bounded and unbounded components. SS covers the bounded demand, while the latter is managed using external measures like expediting shipment or outsourcing. This approach ensures consistent fulfillment times and SLs.

Since this research employs the GSA model, the literature review primarily focuses on GSA-related studies. For a deeper exploration of SSA and its integration with GSA, comprehensive discussions can be found in the works of Wang (2011), Simchi-Levi & Zhao (2012), Eruguz et al. (2016), and de Kok et al. (2018).

Although the GSA model has a decades-long history, recent years have witnessed a surge in research on this topic (Eruguz et al., 2016). Originating from Kimball's work in 1955 and subsequently published in Kimball (1988), GSA is further developed by Simpson (1958) for single-stage inventory systems to identify inventory policies within serial SCs. The solution to this model is obtained by a dynamic programming algorithm devised by Graves (1988). This model is extended to accommodate diverse practical SC scenarios by Graves & Willems (2000, 2005, and 2008), Humair & Willems (2011), Funaki (2012), Moncayo-Martınez & Zhang (2013), Jiang et al. (2016), Aouam & Kumar (2019), Ghadimi et al. (2020), Aouam et al. (2021).

In the context of GSA, the SS placement problem often employs two common inventory policies for each inventory location in an SC. One is the $(R, Q)$ policy, where an order of quantity $Q$ is placed whenever the inventory position is at or below a reorder point $R$. The second is the base stock policy (order-up-to policy), wherein orders are placed at a regular review period to attain a predetermined base stock level. For example, Shenas et al. (2009) apply the $(R, Q)$ policy to a continuous review of two-stage serial SC, formulating models to calculate reorder points and determine upstream inventory. Similarly, Li & Chen (2012) consider a variant of $(R, Q)$ policy known as the echelon $(R, nQ)$ policy for a general serial SC, employing dynamic programming to optimize SC inventory. Li et al. (2013) adopt this approach for an assembly network with a $(nR, Q)$ policy. Chen & Li (2015) explore problems presented by Li & Chen (2012), as well as Li et al. (2013) under various operational scenarios. Although the $(R, Q)$ policy is prevalent in research, it is less common in practical application. In inventory systems, like warehouses or DCs, the base stock policy is normally employed (Jung et al., 2008), especially for periodic review systems where order consolidation can minimize ordering costs. Consequently, many researchers consider the base stock policy in addressing IP challenges, such as Graves & Willems (2000), Kaminsky & Kaya (2008), Funaki (2012), Moncayo-Martínezn & Zhang (2013), Klosterhalfen et al. (2014), Grahl et al. (2016), Graves & Schoenmeyr (2016), Aouam & Kumar (2019), Aouam et al. (2021).

Apart from determining inventory policy, another crucial aspect of the IP problem is the demand distribution assumption. To make the problem analysis simpler, many research studies adopt the assumption that demand conforms to theoretical distributions such as the normal distribution (Graves & Willems, 2000; Humair & Willems, 2008; Kaminsky & Kaya, 2008; Jung et al., 2008; Moncayo-Martínezn & Zhang, 2013; Klosterhalfen et al., 2014; Grahl et al., 2016; Graves & Schoenmeyr, 2016; Aouam & Kumar, 2019; Ghadimi et al., 2020; Aouam et al., 2021) or follows a stochastic process where demand retains a normal distribution with dynamic variance Graves & Willems (2005), Funaki (2012).

Given a particular demand distribution, the total demand during a replenishment cycle within an SC can be categorized into two unequal parts. The larger portion, known as bounded demand, is satisfied by available inventory, while the smaller portion, referred to as unbounded demand, is addressed through operational flexibilities, such as accelerated production Ghadimi et al. (2020) or subcontracting Aouam and Kumar (2019), Aouam et al. (2021). Within this demand-splitting framework, fulfillment timing (or RT) is consistently ensured. Moreover, bounded demand, which constitutes the portion of total demand met by the inventory system in a replenishment cycle, has an implicit influence on the CSL of the SC. As a result, numerous studies focusing on GSA establish the magnitude of bounded demand by defining a CSL, exemplified as 90% (Graves & Willems, 2000), 95% (Funaki, 2012; Moncayo-Martínezn & Zhang, 2013; Kaminsky & Kaya, 2008), 97.5% (Ghadimi et al., 2020; Aouam et al., 2021), or 97.7% (Graves & Schoenmeyr, 2016).

Traditionally, SLs are predefined by customers or managers. Consequently, much of GSA research considers CSLs as fixed inputs, focusing on minimizing total inventory cost while often overlooking the influence of managing additional unbounded demand. However, in practical settings, managers and customers frequently set CSLs based on experience and preference rather than comprehensive evaluations of trade-offs involving factors like operational flexibility and inventory carrying costs (Chen & Li, 2015). In this context, a meticulous evaluation of these factors can result in a superior SL that effectively minimizes total inventory costs (Jung et al., 2008). This approach is demonstrated by the work of Aouam & Kumar (2019), where they prove that optimizing CSLs alongside SS placement decisions and considering supplementary measures, such as subcontracting and overtime, leads to decreased overall SS costs compared to using a predefined SL.

Table 2.1 illustrates the advancements of current research as it concurrently addresses multiple realistic aspects of the SS placement problem, building upon the groundwork of prior studies.

**Table 2.1** Summary of studies on the inventory positioning problem

| Papers | Inventory policy | Demand distribution | CSL | Shortage |
|---|---|---|---|---|
| Graves & Willems (2000) | BS | Normal | F (90%) | |
| Graves & Willems (2005) | BS | Normal distribution with a dynamic variance | F (95%) | |
| Simchi-Levi & Zhao (2005) | BS | Poisson | N | ✓ |
| Kaminsky & Kaya (2008) | BS | Normal | F (95%) | |
| Jung et al. (2008) | BS | Normal | Y | ✓ |
| Funaki (2012) | BS | Normal distribution with a dynamic variance | F (95%) | |
| Li et al. (2013) | $(R, Q)$ | Poisson | F | |
| Moncayo-Martínez & Zhang (2013) | BS | Normal | F (95%) | |
| Chen & Li (2015) | $(R, Q)$ | Poisson | F | |
| Graves & Schoenmeyr (2016) | BS | Normal | F (97.7%) | ✓ |
| Aouam & Kumar (2019) | BS | Normal | Y | ✓ |
| Ghadimi et al. (2020) | BS | Normal | F (97.5%) | |
| Aouam et al. (2021) | BS | Normal | F (95%, 97.5%) | |
| This study | BS | Normal | Y | ✓ |

*BS: Base stock; N: No consideration; F: Fixed cycle service level; Y: Flexible cycle service level*

**2.2 Network design problem**

The NDP is classified into three main categories: forward (Gen et al., 2006; Kaminsky & Kaya, 2008; You & Grossmann, 2008; Altiparmak et al., 2009; Lin et al., 2009; Lotfi & Tavakkoli-Moghaddam, 2013; Rahmaniani & Ghaderi, 2013; Zadeh et al., 2014; Rodriguez et al., 2014; Taxakis & Papadopoulos, 2016; Gen et al., 2018; Savadkoohi et al., 2018; Rohaninejad et al., 2018; Puga et al., 2019; Cortinhal et al., 2019; Shoja et al., 2019, 2020; Durmaz & Bilgen, 2020; Nezamoddini et al., 2020; Robles et al., 2020; Darmawan et al., 2021; Hasani et al., 2021), reverse (Lee et al., 2009; Alumur, 2012; Alshamsi & Diabat, 2017; John et al., 2018; Gholizadeh et al., 2022), and closed-loop SCs (Amin & Zhang, 2013; Zeballos et al., 2014; Cui et al., 2017; Shi et al., 2017; Diabat & Jebali, 2021; Fathollahi-Fard et al., 2020; Ghahremani-Nahr, 2019; Gholizadeh & Fazlollahtabar, 2020; Pazhani et al., 2021).

In this part, we concentrate on reviewing the studies related the forward SC domain. Noteworthy studies within this field include those addressing single-product, single-period scenarios, such as Gen et al. (2006), Kaminsky & Kaya (2008), Lotfi & Tavakkoli-Moghaddam (2013), Gen et al. (2018), Puga et al. (2019), Shoja et al. (2020), Darmawan et al. (2021). Expanding on this, different research delves into various directions: multi-product, single-period (Kaminsky & Kaya, 2008; Taxakis & Papadopoulos, 2016; Cortinhal et al., 2019; Shoja et al., 2019; Durmaz & Bilgen, 2020; Robles et al., 2020); single-product, multi-period (Altiparmak et al., 2009; Lin et al., 2009; Gen et al., 2018; Nezamoddini et al., 2020); and multi-product, multi-period (You & Grossmann, 2008; Zadeh et al., 2013; Rodriguez et al., 2014; Taxakis & Papadopoulos, 2016; Savadkoohi et al., 2018; Durmaz & Bilgen, 2020).

Table 2.2 presents an overview of prior studies relevant to our problem, encompassing elements such as objective functions, number of products, planning horizon, BOM, and other decisions. Among these studies, two standard objective functions are observed: total SC profit maximization (You & Grossmann, 2008; Durmaz & Bilgen, 2020; Hasani et al., 2021) and system-wide cost minimization (You & Grossmann, 2008; Taxakis & Papadopoulos, 2016; Durmaz & Bilgen, 2020). Common cost elements, such as opening/fixed, transportation, inventory, and production/operation costs, are typically considered. Only a limited number of studies delve into uncommon cost components, such as facility expansion capacity (Rodriguez et al., 2014; Durmaz & Bilgen, 2020), shortage (Rodriguez et al., 2014; Taxakis & Papadopoulos, 2016; Savadkoohi et al., 2018), and procurement costs (You & Grossmann, 2008; Taxakis & Papadopoulos, 2016; Durmaz & Bilgen, 2020). Environmental concerns are integrated into a few studies as additional objectives (Robles et al., 2020; Hasani et al., 2021). For instance, Robles et al. (2020) explore a multi-objective model for a hydrogen network. They aim to optimize global warming potential, levelized hydrogen cost, and total risk. Hasani et al. (2021) tackle a multi-objective, multi-period, multi-product SC context. Three objectives are simultaneously optimized, including profit, $CO_2$ emissions, and green facility centralization to mitigate disruption risks.

In terms of the number of products and planning horizon, a majority of studies focus on scenarios involving multiple products and multiple periods. This focus is rooted in the long-term impact of strategic planning and the common practice of diverse product deliveries within SCs. The intricate relationships between products' components, manufacturing methods, sourcing constraints, total costs, production processes, overall throughput capabilities, and capital decisions are encapsulated in a BOM. However, the integration of BOM into SC systems has yet to be widely considered due to the BOM's influence on flow balance constraints. Facility location models (FLP) require a balance between inflows and outflows at each node, assuming homogenous incoming and outgoing flows. However, the incorporation of BOMs into FLP necessitates adaptations and modifications to this equilibrium. Given the complexity associated with BOMs, most studies tend to overlook this aspect in their models (Gen et al., 2006; Lin et al., 2009; Lotfi & Tavakkoli-Moghaddam, 2013; Gen et al., 2018; Nezamoddini et al., 2020; Robles et al., 2020; Shoja et al., 2019 and 2020; Darmawan et al., 2021).

In the past, the BOM is primarily considered with regard to the product return rate within reverse SCs (Lee et al., 2009; Alshamsi & Diabat, 2017; Gholizadeh et al., 2022) or in the context of closed-loop SCs (Cui et al., 2017; Shi et al., 2017; Gholizadeh & Fazlollahtabar, 2020). When it comes to traditional forward SCs, the incorporation of BOM considerations has been explored in just a few studies (Altiparmak et al., 2009; Zadeh et al., 2013; Taxakis & Papadopoulos, 2016; Durmaz & Bilgen, 2020; Hasani et al., 2021). Notably, Zadeh et al. (2013) explore BOM within a steel-making network. Similarly, Durmaz & Bilgen (2020) develop an optimization model that incorporates BOM aspects, particularly conversion rates of biogas and biomass, for the production of electrical energy and digestion in a sustainable biomass SC network. Hasani et al. (2021) investigate BOM in the context of a medical device manufacturing company network in Iran.

**Table 2.2** Summary of relevant studies

| Study | Economic performance (income/expense) | | | | | | | | | Emission/ environmental aspect | Period-Product | BOM | Decision | Safety stock position | Demand |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Revenue | Opening/fixed | Capacity installation | Transportation | Inv. related | SS holding | Production/ operation | Shortage | Procurement | | | | | | |
| You & Grossmann, 2008 | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | | (M,M) | | NDP+IP | DCs | Normal, Triangular |
| Atamtürk et al., 2012 | | ✓ | | ✓ | | ✓ | | | | | (S,S) | | NDP+IP | DCs | Normal |
| Funaki, 2012 | | ✓ | | ✓ | | ✓ | ✓ | | | | (M,S) | | NDP+IP | All positions | Stationary, nonstationary |
| Zadeh et al., 2013 | | ✓ | | ✓ | | ✓ | ✓ | | | | (M,M) | ✓ | NDP+IP | WHs | Normal |
| Rodriguez et al., 2014 | | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | | (M,M) | | NDP+IP | WHs, Customers | Normal, Poisson |
| Taxakis & Papadopoulos, 2016 | | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | (M,M) | | NDP | | |
| Savadkoohi et al., 2018 | | ✓ | | ✓ | ✓ | | | ✓ | | | (M,M) | | NDP | | |
| Puga et al., 2019 | | ✓ | | ✓ | ✓ | ✓ | | | | | (S,S) | | NDP+IP | DCs, Retailers | Normal |
| Durmaz & Bilgen, 2020 | ✓ | | ✓ | | | | ✓ | | ✓ | | (M,M) | ✓ | NDP | | |
| Robles et al., 2020 | | | | ✓ | ✓ | | ✓ | | | ✓ | (M,M) | | NDP | | |
| Hasani et al., 2021 | ✓ | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | (M,M) | ✓ | NDP | | |
| This study | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | (M,M) | ✓ | NDP+IP | All positions | Normal |

***Note:*** *Period-Product: S denotes "Single"; M denotes "Multiple"*
*Decision: NDP denotes "Network Design Problem"; IP denotes "Inventory Positioning Problem"*
*Safety stock position: DC denotes "Distribution Center"; WH denotes "Warehouse"*

## 2.3 The integration of network design and inventory positioning problem

Besides addressing the multi-product, multi-period dimensions of the problem, some studies also incorporate the SS structures as a decision, particularly when faced with uncertain demand, referred to as the IP problem. As mentioned before, decisions regarding IP (Graves & Willems, 2000; Simchi-Levi & Zhao, 2005; Jung et al., 2008; Li et al., 2013; Chen & Li, 2015; Graves & Schoenmeyr, 2016; Aouam & Kumar, 2019; Ghadimi et al., 2020; Aouam et al., 2021) are made independently with the NDP. It is noteworthy that there exists limited research that explicitly integrates these two models (Atamtürk et al., 2012; Funaki, 2012; Zadeh et al., 2013; Puga et al., 2019; Rodriguez et al., 2014; You & Grossmann, 2008).

Specifically, the study conducted by Atamtürk et al. (2012) delve into a comprehensive examination of the facility location-inventory problem across various scenarios. These scenarios encompass situations characterized by limited facility capacity, interdependent demands, and multiple commodities. Their primary contribution is the development of these intricate models as mixed-integer programming, along with conic quadratic constraints. On the other hand, Puga et al. (2019) focus on research within a single-echelon SC, with single-sourcing, single-product, and single-period contexts. Their work is rooted in the $(R, Q)$ inventory policy, where demand at retailers follows a normal distribution that is uncorrelated, with known mean and variance. To tackle this, they formulate a conic quadratic mixed-integer program (CQMIP), which is suggested by Atamtürk et al. (2012).

Another study conducted by Funaki (2012) introduces a systematic approach characterized by a stepwise process, focusing on both the selection of network options and the positioning of SS within the selected option. This approach extends the GSA to incorporate a case with due-date-based demand. In addition, dynamic programming is employed as a crucial tool for determining the optimal placement of SS while concurrently making decisions regarding location selection.

Other studies explore the integration of NDP and IP in the context of multi-period, multi-product SC networks in diverse industrial contexts, such as chemical and steel industries, such as Zadeh et al. (2013), Rodriguez et al. (2014), and You & Grossmann (2008). These studies share a common objective of optimizing the configuration of SC networks employing a periodic review inventory policy while accounting for demand uncertainty across extended planning horizons. In particular, the challenge of determining the optimal placement of SSs is addressed within their optimization models. Despite their common goals, there are differences among these studies. For instance, Rodriguez et al. (2014) only consider the impact of lost sales resulting from shortages, enhancing the realism of their model. Conversely, Zadeh et al. (2013) take into account the BOM, acknowledging the complexity introduced by the interdependencies of products and components within a SC.

These studies collectively contribute to advancing our understanding of the intricate interconnection between ND decisions, IP strategies, and demand uncertainty management in multi-product, multi-period SC contexts.

**2.4 Heuristics approach**

In addressing the NDP, two primary methods are commonly employed: exact techniques using commercial optimization solvers (You & Grossmann, 2008; Rodriguez et al., 2014; Savadkoohi et al., 2018; Cortinhal et al., 2019; Puga et al., 2019) and heuristic approaches (Gen et al., 2006; Altiparmak et al., 2009; Lin et al., 2009; Lotfi & Tavakkoli-Moghaddam, 2013; Taxakis & Papadopoulos, 2016; Alshamsi & Diabat, 2017; Cui et al., 2017; Shi et al., 2017; Gen et al., 2018; Shoja et al., 2019, 2020; Gholizadeh & Fazlollahtabar, 2020; Nezamoddini et al., 2020; Robles et al., 2020; Darmawan et al., 2021; Gholizadeh et al., 2022). While exact methods demonstrate effectiveness for smaller to moderate-scale problems, they become unfeasible for larger, practical scenarios due to the time-consuming nature and complexity of the issues involved (Paithankar & Chatterjee, 2019). To tackle the challenges posed by these larger and more intricate problems, the application of metaheuristic algorithms has gained prominence across diverse domains. In the context of the SCND, which involves the optimization of facility and capacity selection while considering customer demand and cost minimization within an SC network structure, heuristic strategies have demonstrated their efficacy (Altiparmak et al., 2009). These approaches are inspired by natural phenomena, yielding efficient and effective problem-solving methodologies. Metaheuristics such as GAs, simulated annealing, harmony search, and particle swarm optimization have gained significant attention within the optimization community (Kumar et al., 2014).

Metaheuristic algorithms exhibit distinct strengths and weaknesses. To evaluate the performance of different optimization techniques when employed in various combination strategies, El-Beltagy & Keane (1999) conduct a comprehensive comparative analysis. The focus of their study is primarily on evaluating the effectiveness of a GA based on clustering and sharing (GACS) in contrast to 29 alternative methods, such as adaptive random search, evolutionary programming, simulated annealing. They employ a smooth function with multiple peaks of similar heights, referred to as a bump function, as the optimization problem. The outcomes of the study highlight that GACS excels when it disperses the population across numerous peaks in a dynamically changing fitness environment. Additionally, another research introduces a multi-level optimization approach and demonstrates that the GA frequently succeeds in locating the region of the search space that contains the global optimum, as validated by Erbatur et al. (2000). Furthermore, one of the most appealing attributes of the GA is its self-sufficiency throughout the optimization process, without necessitating any additional information. This study concludes that within the field of discrete optimization methodologies, the GA stands out as a robust and promising strategy.

**2.5 Genetic algorithm**

The GA approach was initially introduced by Holland (1975) during the early 1970s, and since then, it has found successful application in diverse combinatorial search space problems. GAs are regarded as stochastic search techniques that simulate natural procedures such as selection, crossover, mutation, locality, and neighborhood, in the metaphor of the concept of natural biological evolution (Thompson, 2016).

In the context of solving the SCNDP, GAs have demonstrated their effectiveness. The GA framework provides a robust means to effectively navigate the intricacies and complexities associated with multi-stage logistics and SC operation management.

The application of GAs involves a critical consideration: the encoding process, which is referred to as the procedure of representing potential solutions to a problem in a form that the algorithm can work with, and the decoding process, which entails converting the solution representation into a feasible solution. Researchers have investigated various encoding/decoding methods within GAs, aiming to interpret and generate feasible solutions. Each encoding technique presents distinct strengths and trade-offs, depending on the specific problem's attributes and the optimization goals. The choice of an appropriate decoding method is significant, as it directly impacts the quality and feasibility of the solutions derived from the GA process.

Alshamsi & Diabat (2017) propose a custom encoding process for their optimization problem. Within this approach, the chromosomes embody binary variables extracted from the MILP model. Each gene within the chromosome is randomly assigned a value of either zero or one, signifying the absence or presence of a specific decision in the solution. This encoding process ensures the efficient generation of feasible solutions by translating the binary chromosome representation into meaningful values for decision variables.

Similarly, Gholizadeh et al. (2022) adopt an encoding strategy that involves the assignment of random values to different segments of the chromosome to address a reverse logistics problem. Each segment of the chromosome corresponds to specific attributes or aspects relevant to the considered problem. Through the random assignment of values to these segments, they can systematically explore and evaluate various combinations and configurations within the framework of the GA, enabling the finding of good solutions for their reverse logistics problem.

For the closed-loop SCND, Cui et al. (2017) develop an encoding scheme where a chromosome conveys comprehensive details about the allocation of sites to intermediate centers, producers, and customer centers. Furthermore, the connections between customer centers and intermediate centers, or between intermediate centers and producers, are indicated within a chromosome. The initial row of the chromosome indicates the allocation of location sites, while the subsequent row signifies the linkages between different facilities. A similar approach based on the concept of random value assignment is also in the works of Gholizadeh & Fazlollahtabar (2020), Nezamoddini et al. (2020), and Robles et al. (2020).

Priority-based encoding is considered one of the most well-known encoding techniques, which is introduced by Gen & Cheng (2000). In this encoding approach, priorities are assigned to different elements or variables of the problem. The chromosome represents a potential solution, where each gene in the chromosome corresponds to a specific component or variable. The value of the gene indicates the priority attributed to the variable. By using priority-based encoding, GAs can effectively explore and optimize solutions based on established priorities, potentially yielding favorable outcomes. The implementation of the priority-based encoding can be found in many NDPs, such as Gen et al. (2006), Altiparmak et al. (2009), Lin et al. (2009), Lee et al. (2009), Lotfi et al. (2013), Tari & Hashemi (2016), Taxakis & Papadopoulos (2016), Shi et al. (2017), Gen et al. (2018), Shoja et al. (2019, 2020), and Darmawan et al. (2021).

Gen et al. (2006) introduce a novel GA featuring priority-based encoding and decoding methods while considering a two-stage transportation problem. Their approach effectively applies priority-based encoding to a single-period, single-product network problem with known maximum capacity. In contrast to Prüfer number encoding, also known as spanning tree-based GA, in transportation problems, the priority encoding method proves more suitable for transportation networks due to its elimination of the need for repair mechanisms, as required by Prüfer number encoding. Altiparmak et al. (2009) extend the priority-based encoding approach to address a complex problem involving single-sourcing, multi-product, and multi-echelon scenarios. In a comparative study against CPLEX, their experimental results highlight that the steady-stage GA outperforms other techniques like Langrangean heuristic and simulated annealing in terms of generating superior heuristic solutions with relatively minimal computational time. Furthermore, Lotfi & Tavakkoli-Moghaddam (2013) apply the priority-based GA method proposed by Gen et al. (2006) to a fixed-charge transportation problem. Through numerical analysis, their study demonstrates a clear advantage of the proposed priority-based GA over the spanning tree-based GA, showcasing better solution quality and computation time. Tari & Hashemi (2016) also use the priority-based GA technique to address a heterogeneous fleet transportation problem. Their formulation incorporates a nonlinear transportation cost function aligned with a vehicle rental discount policy. Similar to previous studies, the results of their investigation point out that the proposed technique is capable of providing good solutions while requiring less computational effort, as compared to CPLEX-based solutions. Taxakis & Papadopoulos (2016) leverage the priority-based encoding GA approach to solve mixed-integer linear and nonlinear programming models. Their research considers multi-period, multi-product, and multi-stage SCNDPs, as well as production distribution and inventory planning problems.

Besides the conventional SCNDPs, Lin et al. (2009) and Gen et al. (2018) introduce a hybrid evolutionary algorithm (EA) to address a multi-stage SC network model integrating the trade-off between indirect and direct distribution. Their study employs a priority-based encoding with extensions, a local search strategy, and a novel fuzzy logic control for enhancing the EA's search capabilities, particularly due to the intricate network structure. Shoja et al. (2019, 2020) utilize a priority-based encoding GA to tackle a flexible SCND challenge involving diverse delivery modes, including both direct and non-direct shipment and delivery. Their objective is to come up with a solution methodology that can handle decision-making complexities associated with designing a flexible SC network. By integrating the priority-based encoding technique, they aim to optimize the selection and configuration of delivery modes to enhance the overall efficiency and performance of the SC network. Darmawan et al. (2021) develop a model for designing a single-product SC network with two echelons while considering the synchronization of reorder points, i.e., SSs, to minimize the total cost while maintaining a high CSL. They devise a heuristic approach based on a GA to tackle the problem's intricacies. In contrast to the conventional forward SC problem, Lee et al. (2009) devise a priority-based GA for a single-period, single-product reverse logistics network challenge. Similarly, Shi et al. (2017) develop a priority-based encoding GA to solve a multi-objective, single-product, single-period closed-loop ND problem. In this study, a predetermined minimum capacity is set to restrict the use of a facility upon opening.

Table 2.3 provides a concise overview of the problems tackled in various studies, along with GA aspects such as hyper-parameters, termination conditions, and tuning choices.

**Table 2.3** Summary of GA configuration

| Study | Echelon | Multi-period | Multi-product | BOM/return rate | Capacity | Genetic Algorithm: Encoding scheme | Selection operator | Population size | Crossover | Mutation | Termination condition | Tuning |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Gen et al. (2006) | ME | | | | F | Priority-based | Roulette wheel selection with elitist strategy | 10, 20, 30, 75 | Partially mapping, Order, Position-based, Weight mapping Prob = 0.5 | Insert, Swap Prob = 0.5 | Number of generations | ✓ |
| Lin et al. (2009) | ME | ✓ | | | F | Priority-based | Local search technique | 100 | Partial-mapped, Two-point crossovers Prob = 0.6, 0.9 | Inversion, Insertion Prob = 0.6, 0.9 | Number of generations | ✓ |
| Altiparmak et al. (2009) | ME | | ✓ | ✓ | F | Priority-based | Binary tournament selection mechanism | 50 | Uniform Prob = 1 | Swap, Conventional Prob = 0.5 | CPU running time | |
| Lee et al. (2009) | RME | | ✓ | ✓ | F | Priority-based, Prüfer number-based | Roulette wheel method | 10 | Weight mapping Prob = 0.7 | Insertion Prob = 0.7 | Number of generations | |
| Lotfi & Tavakkoli-Moghaddam (2013) | SE | | | | N | Priority-based, spanning tree-based | $(\mu + \lambda)$-selection, roulette wheel, and elitist mechanism | 100 | Order of priority exchange, Priority exchange Prob = 0.2 | Priority exchange Prob = 0.4 | Number of generations | |
| Taxakis & Papadopoulos (2016) | ME | ✓ | ✓ | ✓ | F | Priority-based | N/A | 25, 50 | Uniform using binary mask Prob = 0.5 | Swap using binary mask Prob = 0.5 | Comparing the current best $j$ and best $j + 1 - l$ | |
| Shi et al. (2017) | C | | | ✓ | F | Priority-based | Binary tournament selection strategy | 200 | Uniform using binary mask Prob = 0.25 | Swap using binary mask Prob = 0.5 | Number of generations | |
| Alshamsi & Diabat (2017) | RME | | ✓ | ✓ | F | The chromosomes are represented in binary | Elitism, tournament selection | 20, 40, 60, 80 | Multi-bit Prob = 0.7 | Single-bit Prob = 0.05 | Number of generations | |
| Cui et al. (2017) | C | | | ✓ | F | Each gene represents the location points and the connections between different facilities | Tournament selection | 200, 300, 400, 500 | Order, Two-point | Swap | Number of generations | ✓ |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Gen et al. (2018) | ME | | | | F | Priority-based | Roulette wheel method | 400 | Partial-mapped, Two-point crossovers Prob = 0.5 | Insertion Prob = 0.7 | Number of generations | ✓ |
| Shoja et al. (2019) | ME | | ✓ | | F | Priority-based | N/A | 60, 70, 80, 90, 100, 110, 120 | One-point, Two-point, Uniform, Arithmetic Prob = 0.75, 0.8, 0.85, 0.9 | Swap, Big swap, Displayment, Inversion, Scramble, Insertion, Modified boundary, Random minor, Random part Prob = 0.15, 0.2, 0.25, 0.3 | Number of generations | ✓ |
| Gholizadeh & Fazlollahtabar (2020) | C | ✓ | ✓ | ✓ | F | A sequence of real random numbers between zero and one | Best individuals are selected | 250 | One-point Prob = 0.9 | Inversion with local search Prob = 0.1 | Number of generations | |
| Shoja et al. (2020) | ME | | | | F | Priority-based | N/A | [60, 130] | One-point, Two-point, Uniform, Arithmetic Prob = 0.6-0.9 | Swap, Big swap, Displayment, Inversion, Scramble, Insertion, Modified Boundary, Random minor, Random part, Prob = 0.1-0.4 | Max iteration CPU running time | ✓ |
| Robles et al. (2020) | ME | | ✓ | | F | The chromosome of the variables is complemented by a vector containing the type of variable | Binary tournament selection mechanism | 500, 2000 | Simulated binary Prob = 0.9 | Parametric Prob = 0.5 | Number of generations | |
| Nezamoddini et al. (2020) | ME | ✓ | | | F | Each gene represents a proportion of the available flow in the source nodes | Roulette wheel method | 10 | Multi-point Prob = 1 | Node, link, and source mutation Prob = 0.01, 0.1, 0.3 | Number of generations | ✓ |
| Darmawan et al. (2021) | ME | | | | F | Priority-based | Select the best individual, roulette wheel, random generating | 100 | One-point Prob = 0.8 | Swap Prob = 0.2 | Number of generations No improvement found in 150 consecutive iterations | |
| Gholizadeh et al. (2022) | RME | ✓ | ✓ | ✓ | F | A chromosome consists of multiple parts, each gene in the part holds an integer | Tournament selection | 100, 170, 250 | Modified one-point Prob = 0.6, 0.7, 0.8 | One-point Prob = 0.1, 0.2, 0.3 | Number of generations | ✓ |
| This study | ME | ✓ | ✓ | ✓ | M | Rank-based | Select the best individual, roulette wheel | 20, 40 | One-point, Order Prob = 1 | Inversion, Swap Prob = 0.1, 0.2, 0.3 | Number of generations | ✓ |

***Echelon**: SE: Single-echelon, ME: Multi-echelon, RME: Reverse Multi-echelon, C: Closed-loop*
***Capacity**: N: Not given, F: Fixed capacity, M: Flexible capacity (can be expanded)*

**2.6 Research gap**

It should be noted that the two studies of Atamtürk et al. (2012) and Puga et al. (2019), despite their significance, solely explore the problem within a single period and single-product context. Furthermore, while both studies incorporate IP, involving the determination of SS quantities and placements, the shortage is not considered in their analyses.

On the other hand, the study conducted by Funaki (2012) also has certain limitations. Specifically, their research scope is confined to analyzing an assembly SC configuration characterized by a single product and the presence of only one final assembly node. Although their work provides valuable insights into integrating ND decisions and IP using dynamic programming, the broader applicability of their findings could be constrained when dealing with more intricate SC configurations that involve multiple products or complex network structures.

In addition, the specific facilities designated for storing SSs are limited to certain locations, such as warehouses (Zadeh et al., 2013; Rodriguez et al., 2014) or distribution centers (You & Grossmann, 2008). However, it is important to note that this approach might not cover all possible strategies for SS placement. In more complex SCs or industries, SS might need to be strategically positioned at other points, such as manufacturing plants, intermediate nodes, or even retail locations, depending on the network structure and operational considerations.

As far as we can tell, there is one study (Darmawan et al., 2021) that employs a heuristic approach, specifically a GA, to address the complexity of the integrated problem. However, it should be noted that Darmawan et al. (2021) focus on a SCND problem with fill rate as the key service performance metric. In addition, it is crucial to highlight that Darmawan et al. (2021) only concentrate on single-period, single-product SC.

In contrast to the mentioned papers, our study takes a distinct approach by integrating the strategic decision-making process of multi-echelon production ND with the tactical decision of IP. We adopt the GSA to formulate the IP model, ensuring that customer demand is met with a specified SL. Furthermore, our model accounts for the BOM, a crucial aspect given of a production network. Consequently, the determination of SS amounts is based on the product profile.

Another key consideration in our study is the incorporation of varying CSLs, empowering the model to recommend the optimal CSL that yields the lowest overall cost. Moreover, unlike prior research that confines SS placement to specific entities like warehouses (Rodriguez et al., 2014) or DCs (You & Grossmann, 2008; Atamtürk et al., 2012), our model offers the flexibility to position SS at any location within the network.

Dissimilarly to previous studies that address non-linear integer programming models for IP through techniques like linear relaxation (Rodriguez et al., 2014) or hierarchical algorithms (You & Grossmann, 2008), our research proposes a unique method to linearize non-linear components without relaxation. This innovative approach enables us to construct a MILP model that not only delivers the optimal solution but also significantly enhances computational efficiency.

When considering the heuristic approach for addressing large-scale problems, our study diverges from Darmawan et al. (2021) in terms of focusing on cost optimization rather than fill rate optimization. Our approach to SS management also differs significantly. While Darmawan et al. (2021) employ coordinated reorder points and fixed batch quantities, we adopt a distinct strategy based on a base stock policy that considers normally distributed demand. Through this methodology, we aim to optimize SS levels while accounting for demand uncertainty. Additionally, our study expands its scope to encompass a more complex multi-period, multi-product scenario.

Through systematic evaluation and validation via a numerical experiment, we comprehensively assess these integrated features. The outcomes of the experiment not only offer valuable insights but also demonstrate the practical feasibility and robustness of these integrated aspects in addressing challenges within SCNDPs.

# CHAPTER 3
# A MATHEMATICAL MODEL FOR THE INTEGRATED PROBLEM OF PRODUCTION NETWORK DESIGN AND INVENTORY POSITIONING

## 3.1 Problem statement

In this research, we consider an integrated problem encompassing multi-period, multi-echelon ND and IP for a production company. The network configuration involves various facilities, each with specific operational characteristics, interconnected by arcs representing material flows. To capture the material requirements of finished products, we employ BOM. The integrated MILP model is developed based on the context of a single-sourcing strategy, where each component or raw material is supplied exclusively by a single supplier. Furthermore, our investigation delves into a finite planning horizon that is discretely divided into intervals. The scope of the production NDP pertains to multiple periods wherein costs, supply capacity, and demand fluctuate. Within this framework, the final downstream facility's demand is pre-determined and must be satisfied. All cost parameters, consisting of fixed facility opening, fixed capacity installation, unit holding, transportation, and production costs across each period, are considered deterministic.

In addressing the IP problem, we adopt the assumption that the RT connecting any two nodes is identified by the quoted service time from the upstream nodes and the processing time at the current node. Within our model, the processing time at each node is deterministic and remains unaffected by the order quantity, as outlined by Duc et al. (2022). In addition, if a downstream node necessitates various components sourced from multiple upstream nodes, the downstream node must wait until all the required components have arrived before initiating processing. Conversely, when a single upstream node supplies multiple downstream nodes, we assume that the upstream node can provide different service times to respective customers.

The conventional approach to solving the ND and IP problems normally addresses them sequentially and independently. Typically, the ND needs to be solved before the inventory positions can be determined. In our research, we refer to this conventional approach as a two-stage approach, also known as the sequential approach. This approach entails solving the ND using one model and then employing its outcomes as input for the subsequent IP model. It should be noted that the two-stage approach might overlook potential interdependencies between these two problems. In response to this potential limitation, we propose an integrated approach that addresses both problems simultaneously. In our problem formulation, the following notations are employed to denote various variables and parameters.

**Sets and Indices**

$L_1$: Set of nodes on layer 1, component manufacturers

$L_2$: Set of nodes on layer 2, subassembly manufacturers

$L$: A two-dimensional set, $L = L_1 \times L_2$

$K$: Set of final assembly manufacturers,

$LK$: A two-dimensional set, $LK = L_2 \times K$

$N$: Set of nodes indexed by $i, j, k \in N$, where $N = L_1 \cup L_2 \cup K$,

$C$: Set of components of set $L_1$, indexed by $c \in C$,

$SA$: Set of subassemblies of set $L_2$, indexed by $a \in SA$,

$P$: Set of finished products of set $K$, indexed by $p$, $p \in P$

$T$: Set of periods within the planning horizon, $t \in T$,

$W$: Set of capacity levels, $w \in W$,

$BOMC$: BOM, showing the number of components $c$ needed to produce one unit of subassembly $a$, $(c, a) \in BOMC$,

$BOMA$: BOM, showing the number of subassembly $a$ needed to produce one unit of finished product $p$ $(a, p) \in BOMA$,

$M$: Set of discrete duration of RT indexed by $m \in M$,

## 3.2 Network design problem

### 3.2.1 Parameters

$D_{p,k,t}$: Demand of finished product $p \in P$ at the final assembly manufacturer $k \in K$ in period $t \in T$;

$BMC_{c,a}$: Amount of component $c \in C$ used to produce subassembly $a \in SA$;

$BMA_{a,p}$: Amount of subassembly $a \in SA$ used to produce finished product $p \in P$;

$CAPC_{w,i}$: Capacity level $w \in W$ of component manufacturer $i \in L_1$;

$CAPA_{w,j}$: Capacity level $w \in W$ of subassembly manufacturer $j \in L_2$;

$HC_{c,i,t}$: Unit holding cost of component $c \in C$ at component manufacturer $i \in L_1$ in period $t \in T$;

$HA_{a,j,t}$: Unit holding cost of subassembly $a \in SA$ at subassembly manufacturer $j \in L_2$ in period $t \in T$;

$SC_{i,j,t}$: Unit transportation cost from component manufacturer $i \in L_1$ to subassembly manufacturer $j \in L_2$ in period $t \in T$;

$SA_{j,k,t}$: Unit transportation cost from subassembly manufacturer $j \in L_2$ to final assembly manufacturer $k \in K$ in period $t \in T$;

$FC_{i,t}$: The fixed opening cost of component manufacturer $i \in L_1$ in period $t \in T$;

$FA_{j,t}$: The fixed opening cost of subassembly manufacturer $j \in L_2$ in period $t \in T$;

$FIC_{w,i,t}$: The fixed cost associated with installing a specific capacity level $w \in W$ at component manufacturer $i \in L_1$ in period $t \in T$;

$FIA_{w,j,t}$: The fixed cost associated with installing a specific capacity level $w \in W$ at subassembly manufacturer $j \in L_2$ in period $t \in T$;

$PC_{c,i,t}$: Unit production management cost of component $c \in C$ at component manufacturer $i \in L_1$ in period $t \in T$;

$PA_{a,j,t}$: Unit production management cost of subassembly $a \in SA$ at subassembly manufacturer $j \in L_2$ in period $t \in T$;

$U$: Minimum utilization rate of an opened production capacity;

$CAP_0$: Initial capacity of facilities;

$TRC_{c,i}$: Inventory turnover rate of component $c \in C$ produced by component manufacturer $i \in L_1$;

$TRA_{a,j}$: Inventory turnover rate of subassembly $a \in SA$ produced by subassembly manufacturer $j \in L_2$.

### 3.2.2 Decision variables

$XC_{c,i,j,t}$: Flow of component $c \in C$ delivered from component manufacturer $i \in L_1$ to subassembly manufacturer $j \in L_2$ in period $t \in T$;

$XA_{a,j,k,t}$: Flow amount of subassembly $a \in SA$ shipped from subassembly manufacturer $j \in L_2$ to final assembly manufacturer $k \in K$ in period $t \in T$;

$YC_{c,i,t}$: Production quantity of component $c \in C$ at component manufacturer $i \in L_1$ in period $t \in T$;

$YA_{a,j,t}$: Production quantity of subassembly $a \in SA$ at subassembly manufacturer $j \in L_2$ in period $t \in T$;

$IC_{c,i,t}$: Cycle inventory of component $c \in C$ stored at component manufacturer $i \in L_1$ in period $t \in T$;

$IA_{a,j,t}$: Cycle inventory of subassembly $a \in SA$ stored at subassembly manufacturer $j \in L_2$ in period $t \in T$;

$ZC_{i,t}$: Binary variable, 1 if component manufacturer $i \in L_1$ is opened in period $t \in T$, 0 otherwise;

$ZA_{j,t}$: Binary variable, 1 if subassembly manufacturer $j \in L_2$ is opened in period $t \in T$, 0 otherwise;

$AC_{i,w,t}$: Binary variable, 1 if an additional capacity of level $w \in W$ is installed at component manufacturer $i \in L_1$ in period $t \in T$, 0 otherwise;

$AA_{j,w,t}$: Binary variable, 1 if an additional capacity of level $w \in W$ is installed at subassembly manufacturer $j \in L_2$ in period $t \in T$, 0 otherwise;

$CC_{c,i,j,t}$: Binary variable, 1 if component $c \in C$ is carried on arc $(i,j) \in L$ in period $t \in T$, 0 otherwise;

$CA_{a,j,k,t}$: Binary variable, 1 if subassembly $a \in SA$ is carried on arc $(j,k) \in LK$ in period $t \in T$, 0 otherwise;

### 3.2.3 Model assumptions

The problem formulation involves the following assumptions:

- Once a facility is established, it will remain operational for the entire planning horizon since the decision to select and set up a facility is strategic and resource-intensive, making subsequent changes or closures impractical.
- Each opened facility possesses an initial capacity, which can be increased by adding additional capacity among specific available levels.
- During each planning period, a facility is permitted to choose only one capacity level for capacity expansion.
- There's a requirement for a minimum utilization rate for every opened facility.
- The model allows for a certain level of excess cycle inventory to be held at each operational facility, which helps meet future demand.
- The demand locations are assumed to be geographically diverse such that the correlation among them is neglectable (independent demands).

To ensure proper network balance and simultaneously minimize the total cost related to facilities, transportation, and inventory, the problem incorporates three key variables: product flow, production quantity, and inventory quantity. These variables contribute to achieving an optimal solution while adhering to the given constraints.

### 3.2.4 Network design MILP model

The model includes multiple cost elements that play a role in the ND segment. The objective is to minimize a combination of transportation, production management, cycle inventory holding, fixed facility opening, and fixed capacity installation costs. The total network cost is mathematically formulated as follows:

$$\begin{aligned} Minimize\ Z_1 = & \sum_{c \in C} \sum_{i \in L_1} \sum_{j \in L_2} \sum_{t \in T} SC_{i,j,t} XC_{c,i,j,t} \\ & + \sum_{a \in SA} \sum_{j \in L_2} \sum_{k \in K} \sum_{t \in T} SA_{j,k,t} XA_{a,j,k,t} \\ & + \sum_{c \in C} \sum_{i \in L_1} \sum_{t \in T} PC_{c,i,t} YC_{c,i,t} + \sum_{a \in SA} \sum_{j \in L_2} \sum_{t \in T} PA_{a,j,t} YA_{a,j,t} \\ & + \sum_{c \in C} \sum_{i \in L_1} \sum_{t \in T} HC_{c,i,t} \left( IC_{c,i,t} + \frac{\sum_{j \in L_2} XC_{c,i,j,t}}{TRC_{c,i}} \right) \end{aligned} \qquad (3.1)$$

$$+\sum_{a\in SA}\sum_{j\in L_2}\sum_{t\in T} HA_{a,j,t}\left(IA_{a,j,t}+\frac{\sum_{k\in K} XA_{a,j,k,t}}{TRA_{a,j}}\right)$$
$$+\sum_{i\in L_1} FC_{i,1}ZC_{i,1}+\sum_{i\in L_1}\sum_{t\in T\setminus\{1\}} FC_{i,t}\left(ZC_{i,t}-ZC_{i,t-1}\right)$$
$$+\sum_{j\in L_2} FA_{j,1}ZA_{j,1}+\sum_{j\in L_2}\sum_{t\in T\setminus\{1\}} FA_{j,t}\left(ZA_{j,t}-ZA_{j,t-1}\right)$$
$$+\sum_{i\in L_1}\sum_{t\in T} FIC_{i,w,t}AC_{i,w,t}+\sum_{j\in L_2}\sum_{t\in T} FIA_{j,w,t}AA_{j,w,t}$$

The total cycle inventory in a period is calculated as the total of inventory, $IC_{c,i,t}$, $IA_{a,j,t}$ and the average inventory in the current period, $\frac{\sum_{j\in L_2} XC_{c,i,j,t}}{TRC_{c,i}}$, and $\frac{\sum_{k\in D} XA_{a,j,k,t}}{TRA_{a,j}}$ multiplied by the inventory cost per unit, $HC_{c,i,t}$, and $HA_{a,j,t}$, respectively.

Subject to

$$\sum_{j\in L_2} XA_{a,j,k,t} \geq \sum_{p\in P}\left(BMA_{a,p}D_{p,k,t}\right) \qquad \forall a\in SA, k\in K, t\in T \tag{3.2}$$

$$\sum_{i\in L_1} XC_{c,i,j,t} = \sum_{a\in SA}\left(BMC_{c,a}YA_{a,j,t}\right)\ \forall c\in C, j\in L_2, t\in T \tag{3.3}$$

$$YC_{c,i,t}-\sum_{j\in L_2} XC_{c,i,j,t} = IC_{c,i,t} \qquad \forall c\in C, i\in L_1, t=1 \tag{3.4}$$

$$YA_{a,j,t}-\sum_{k\in D} XA_{a,j,k,t} = IA_{a,j,t} \qquad \forall a\in SA, j\in L_2, t=1 \tag{3.5}$$

$$YC_{c,i,t}+IC_{c,i,t-1}-\sum_{j\in L_2} XC_{c,i,j,t} = IC_{c,i,t} \qquad \forall c\in C, i\in L_1, t\in T\setminus\{1\} \tag{3.6}$$

$$\text{“}YA_{a,j,t}+IA_{a,j,t-1}-\sum_{k\in D} XA_{a,j,k,t} = IA_{a,j,t} \qquad \forall a\in SA, j\in L_2, t\in T\setminus\{1\} \tag{3.7}$$

$$\sum_{c\in C} YC_{c,i,t} \leq CAP_0 ZC_{i,t}+\sum_{w\in W}\sum_{\tau\in\{1,\dots,t\}} CAPC_{i,w}AC_{i,w,\tau}\ \forall i\in L_1, t\in T \tag{3.8}$$

$$\sum_{a\in SA} YA_{a,j,t} \leq CAP_0 ZA_{j,t}+\sum_{w\in W}\sum_{\tau\in\{1,\dots,t\}} CAPA_{j,w}AA_{j,w,\tau}\ \forall j\in L_2, t\in T \tag{3.9}$$

$$\sum_{c\in C} YC_{c,i,t} \geq U\left(CAP_0 ZC_{i,t}+\sum_{w\in W}\sum_{\tau\in\{1,\dots,t\}} CAPC_{i,w}AC_{i,w,\tau}\right)\ \forall i\in L_1, t\in T \tag{3.10}$$

$$\sum_{a\in SA} YA_{a,j,t} \geq U\left(CAP_0 ZA_{j,t} + \sum_{w\in W}\sum_{\tau\in\{1,\dots,t\}} CAPA_{j,w} AA_{j,w,\tau}\right) \quad \forall j \in L_2, t \in T \tag{3.11}$$

$$\sum_{w\in W} AC_{i,w,t} \leq ZC_{i,t} \qquad \forall i \in L_1, t \in T \tag{3.12}$$

$$\sum_{w\in W} AA_{j,w,t} \leq ZA_{j,t} \qquad \forall j \in L_2, t \in T \tag{3.13}$$

$$ZC_{i,t} \geq ZC_{i,t-1} \qquad \forall i \in L_1, t \in T\setminus\{1\} \tag{3.14}$$

$$ZA_{j,t} \geq ZA_{j,t-1} \qquad \forall j \in L_2, t \in T\setminus\{1\} \tag{3.15}$$

$$\sum_{i\in L_1} CC_{c,i,j,t} = 1 \quad \forall c \in C,\ j \in L_2, t \in T \tag{3.16}$$

$$\sum_{j\in L_2} CA_{a,j,k,t} = 1 \quad \forall a \in SA,\ k \in K, t \in T \tag{3.17}$$

$$XC_{c,i,j,t} \leq CC_{c,i,j,t} BIG \quad \forall c \in C, i \in L_1, j \in L_2, t \in T \tag{3.18}$$

$$XA_{a,j,k,t} \leq CA_{a,j,k,t} BIG \quad \forall a \in SA, j \in L_2, k \in K, t \in T \tag{3.19}$$

$$XC_{c,i,j,t}, XA_{a,j,k,t}, YC_{c,i,t}, YA_{a,j,t}, IC_{c,i,t}, IA_{a,j,t} \geq 0,\ \forall c \in C, a \in SA, i \in L_1, j \in L_2, k \in K, t \in T \tag{3.20}$$

$$ZC_{i,t}, ZA_{j,t}, AC_{i,w,t}, AA_{j,w,t}, CC_{c,i,j,t}, CA_{a,j,k,t} \in \{0,1\}, \quad \forall c \in C, a \in SA, i \in L_1, j \in L_2, k \in K, t \in T \tag{3.21}$$

Constraints (3.2) ensure that the customer demand must be fulfilled entirely. Constraints (3.3) are dedicated to maintaining the balance between the inflow and outflow of items, taking into account the consumption within the BOM. In addition, Constraints (3.4) - (3.7) determine the flows and inventories of products/components at each facility for the current period. Constraints (3.8) and (3.9) are to enforce a production limit at each facility, making sure that the production does not exceed its capacity. Facilities are provided with a base capacity when they are opened and can be expanded over time or upon initial opening. The minimum utilization rate of the capacity at an operational facility is controlled by Constraints (3.10) and (3.11). Constraints (3.12) and (3.13) allow capacity expansion solely for facilities that have been opened. Through Constraints (3.14) and (3.15), continuity of facility operations throughout the entire planning horizon is guaranteed after a facility has been opened. The single-sourcing strategy is established by Constraints (3.16) - (3.19), ensuring that a downstream facility is linked to only one upstream facility. Finally, Constraints (3.20) and (3.21) represent binary decision variables and non-negativity requirements.

### 3.3 Inventory positioning problem

In this section, a MILP model based on GSA is introduced to address an IP problem within an SC operating under the base stock inventory policy. We begin by outlining a straightforward serial SC configuration, wherein products or materials progress sequentially from their source as raw materials to the final consumer node. The simple model centers on a single period and an assembly network encompassing only one final node responsible for producing a finished product. Subsequently, we extend the IP model by involving a production network featuring numerous final manufacturing nodes with multiple products and periods. To facilitate our discussion, the term "item" is used to refer to components, subassemblies, and final products.

#### 3.3.1 The inventory positioning problem for a serial network

##### 3.3.1.1 The mechanism of the IP problem of single product, single final node

Within the SC network, every manufacturing node has the flexibility to maintain two possible stocking locations. The initial location serves as storage for input items received from an upstream node, facilitating their preparation before entering the production process at the given node. Subsequently, the second location serves as storage for the output item once manufactured at the node.

To maintain an appropriate base stock level, at a given stage $j \in N$, it initiates orders for the items it requires and dispatches them to preceding stages immediately upon receiving a customer order. Typically, a period, denoted as $\mathcal{S}_j^{in}$, is necessary for orders from the upstream stages to reach node $j$, while an interval $\mathcal{P}_j$ is required to produce item $j$ for the customer. Since node $j$ commits to fulfilling customer demands after an interval of $\mathcal{S}_j^{out}$ periods, it necessitates a RT denoted as $m$ (where $m \geq 0$) to meet the customer order.

Furthermore, aside from maintaining the SS of the output item, each internal node $j$ has the option to retain a certain level of SS for the input item sourced from an upstream node $i$. This inventory acts as a buffer, serving to reduce the overall supply lead time. The decision to maintain this SS depends on the impact of $\mathcal{S}_i^{out}$ and the transit time from node $i$ to node $j$ ($\mathcal{O}_{i,j}$). This leads us to the formulation of the RT for both scenarios, as expressed by the following equation:

$$m = \begin{cases} \mathcal{S}_j^{in} + \mathcal{P}_j - \mathcal{S}_j^{out} & if \text{ node j keeps output item} \\ \mathcal{S}_i^{out} + \mathcal{O}_{i,j} - \mathcal{S}_j^{in} & if \text{ node j keeps input item} \end{cases} \tag{3.22}$$

Stage $j$ encounters a demand characterized by a mean of $\mu_D$ and a standard deviation (SD) of $\sigma_D$. Consequently, over a span of $m$ periods, the cumulative demand exhibits an average of $\mu_D m$ and an SD of $\sigma_D\sqrt{m}$.

When considering the normal distribution demand, the base stock level $B$ for an item, whether it is the output from stage $j$ or an input item from node $i$—is commonly defined as follows:

$$B = \mu_D m + \Phi^{-1}(s)\sigma_D\sqrt{m} \tag{3.23}$$

where $\Phi^{-1}(s)$ is the inverse of the standard normal cumulative distribution function associated with the CSL of $s$ in a replenishment cycle. Based on the specified base stock level, on average, a fraction $(1-s)$ of the demand during the lead time cannot be fulfilled by the inventory on hand. This amount is assumed to be backlogged, and its expected value is estimated as:

$$\mathcal{LS}_{j,m} = \mathcal{L}(s)\sigma_D\sqrt{m} \tag{3.24}$$

**3.3.1.2 The mathematical model of IP for single product, single final node**

**Sets and indices:**

$N$: set of nodes $i, j$, that represent SC nodes, $i, j \in N = \{1,2, \dots K\}$, where $K$ is the final node closest to the customer;

$ARC$: set of arcs connecting a predecessor node $i$ and a successor node $j$, $(i, j) \in ARC$;

$M$: set of options for the length of RT, $M = \{1, 2, \dots, |M|\}$;

**Parameters:**

$\mu_D$: average customer demand $D$ per period;

$\sigma_D$: SD of demand $D$ per period;

$\Phi^{-1}(s)$: the inverse of standard normal cumulative distribution function associated with a given CSL of $s$ in a replenishment cycle;

$\mathcal{L}(s)$: the standard normal loss function associated with a CSL value of $s$;

$\mathcal{O}_{i,j}$: transportation time from node $i$ to node $j$, $\forall(i,j) \in A$;

$\mathcal{P}_j$: processing time of node $j \in N$;

$\mathcal{R}$: service time that the final node commits to end customers;

$\mathcal{Q}_m$: the known amount of SS to cover demand during a given RT of $m$;

$\mathcal{LS}_m$: the expected amount of shortage within a replenishment cycle when holding $m$ days of SS.

$\mathcal{H}_j$: inventory holding cost per unit per year of the output item of node $j \in N$.

$\mathcal{L}_j$: shortage cost per unit at node $j \in N$.

**Decision Variables:**

$\mathcal{S}_j^{in}$: Incoming service time of node $j \in N$;

$\mathcal{S}_j^{out}$: Outgoing service time of node $j \in N$;

$\mathcal{NO}_j$: RT of node $j \in N$ keeping output item;

$\mathcal{NI}_{i,j}$: RT of node $j \in N$ which requires input item from node $i \in N$;

An IP problem aims to minimize the overall SS cost for both input and output items while achieving a specified CSL. This optimization process involves determining the optimal allocation of SS and its quantity across the SC network. The MILP model for the IP problem is formulated as follows.

Minimize

$$\sum_{j \in N} H_j\, \Phi^{-1}(s)\sigma_D \sqrt{\mathcal{NO}_j} + \sum_{j \in N|(i,j) \in \text{ARC}} H_j\, \Phi^{-1}(s)\sigma_D \sqrt{\mathcal{NI}_{i,j}} \tag{3.25}$$

Subject to

$$\mathcal{S}_i^{out} + T_{i,j} - \mathcal{S}_j^{in} = \mathcal{NI}_{i,j} \quad \forall\, (j,i) \in A \tag{3.26}$$

$$\mathcal{S}_j^{in} - \mathcal{S}_j^{out} + P_i = \mathcal{NO}_j \quad \forall\, i \in I \tag{3.27}$$

$$\mathcal{S}_K^{out} \leq \mathcal{R} \tag{3.28}$$

$$\mathcal{S}_j^{in} \leq |M| \quad \forall\, j \in N \tag{3.29}$$

$$\mathcal{S}_j^{out} \leq |M| \quad \forall\, j \in N \tag{3.30}$$

$$\mathcal{S}_j^{in}, \mathcal{S}_j^{out}, \mathcal{NI}_{i,j}, \mathcal{NO}_j \geq 0 \quad \forall\, i,j \in N \tag{3.31}$$

The objective is to minimize the total cost of the SC for both input and output items across all nodes. Constraints (3.26) and (3.27) determine the RT for each inventory of input and output items, respectively. Constraint (3.28) ensures that the outgoing service time at Node $K$ would not exceed the minimum service time committed to the end customers. Constraints (3.29) and (3.30) represent the selected incoming and outgoing service times bounded by $|M|$. Constraints (3.31) represent the non-negativity of the lead time of different SC nodes.

### 3.3.2 The inventory positioning problem for an extended problem

#### 3.3.2.1 The mechanism of the extended IP problem

When a given upstream node is responsible for supplying multiple downstream nodes, it can commit different fulfillment times for each downstream node. These fulfillment time commitments are referred to as outgoing service times, denoted as $\mathcal{SC}_{c,i,j,t}^{out}, c \in C, i \in L_1, j \in L_2, t \in T$ $(or\ \mathcal{SA}_{a,j,k,t}^{out}, a \in SA,\ j \in L_2, k \in K, t \in T)$. Additionally, the outgoing service times for facilities located at the final downstream nodes, denoted as $\mathcal{SP}_{p,k,t}^{out}, p \in P, k \in K, t \in T$, must not exceed a pre-specified service time limit, i.e., $\mathcal{SP}_{p,k,t}^{out} \leq \mathcal{R}_{k,t}$ $(\forall k \in K)$.

When a downstream node $j$ (or $k$) necessitates input items from its upstream node $i$ (or $j$), it is compelled to wait until all the required input items are delivered. This waiting time is denoted as the incoming service time denoted as $\mathcal{SC}_{c,i,j,t}^{in}$ (or $\mathcal{SA}_{a,j,k,t}^{in}$) for node $j$ (or $k$) to successfully fulfill the order. After receiving the necessary input items, it takes node $j$ (or $k$) a period of $\mathcal{PA}_j$ (or $\mathcal{PP}_k$) to conduct the manufacturing process. Moreover, as part of its commitments, node $j$ (or $k$) pledges to deliver the output items to its downstream node $k$ (or customers) after $\mathcal{SA}_{a,j,k,t}^{out}$ $(or\ \mathcal{SP}_{p,k,t}^{out})$ periods.

Considering the assumptions above, it is plausible that node $j$ (or $k$) may need to hold an SS of either input or output items. Notably, the need for SS arises when the time required to fulfill an order surpasses the outgoing service time.

Mathematically, this RT is formulated as follows:

$$m = \begin{cases} \begin{cases} -\mathcal{SC}_{c,i,j,t}^{out} + \mathcal{PC}_i & \text{if node } i \text{ keeps output items} \\ \mathcal{SC}_{c,i,j,t}^{in} - \mathcal{SA}_{a,j,k,t}^{out} + \mathcal{PA}_j & \text{if node } j \text{ keeps output items} \\ \mathcal{SA}_{a,j,k,t}^{in} - \mathcal{SP}_{p,k,t}^{out} + \mathcal{PP}_k & \text{if node } k \text{ keeps output items} \end{cases} \\ \begin{cases} \mathcal{SC}_{c,i,j,t}^{out} - \mathcal{SC}_{c,i,j,t}^{in} + \mathcal{OC}_{i,j} & \text{if node } j \text{ keeps input items} \\ \mathcal{SA}_{a,j,k,t}^{out} - \mathcal{SA}_{a,j,k,t}^{in} + \mathcal{OA}_{j,k} & \text{if node } k \text{ keeps input items} \end{cases} \end{cases} \quad (3.32)$$

It is assumed that the production times $\mathcal{PC}_i, \mathcal{PA}_j, \mathcal{PP}_k$ are non-negative values. Additionally, the transportation times, $\mathcal{OC}_{i,j}, \mathcal{OA}_{j,k}$ are independent of the production quantity (Simpson, 1958).

#### 3.3.2.2 Standard deviation computation

Apart from the RT, another determinant of safety inventory levels is the fluctuation in demand for a particular item. This variability is transmitted across the SC network, originating from the final downstream nodes, where demand patterns are observed. Given a network configuration, the manufacturing nodes encounter varying degrees of demand variability, quantified by SDs. Although computing these SDs is essential for determining the SS, the process is intricate and complex. To illustrate the calculation process, a small example is presented.

Consider the network including seven nodes, i.e., $L_1 = \{1, 2\}, L_2 = \{3, 4\}, K = \{5, 6, 7\}$, with the flows as illustrated in Figure 3.1 Simple SC with pre-defined connections.

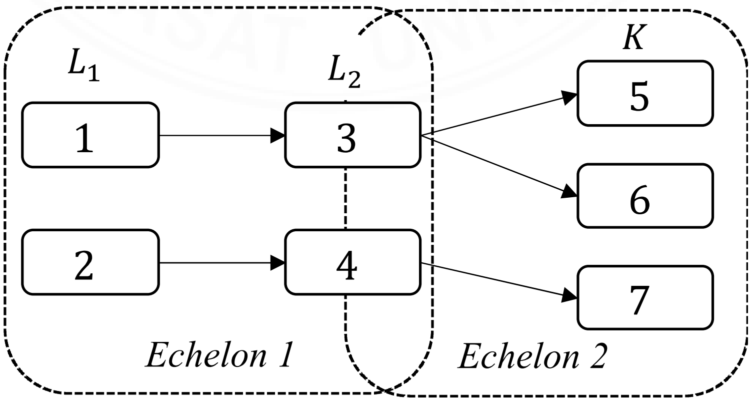



**Figure 3.1** Simple SC with pre-defined connections

The demand SDs for item $p$ at nodes 5, 6, and 7 are given by: $\sigma_{p,5} = 50, \sigma_{p,6} = 75$, and $\sigma_{p,7} = 100$.

Assuming that to manufacture one unit of output item $p$, it is required $\alpha = 1$ unit of subassembly $a$ from the location set $L_2$. Due to the single-sourcing strategy, each downstream node $k \in K$ can only receive items from a single upstream node $j \in L_2$. Consequently, the calculation of the demand SD $\sigma_{a,j,k,t}$ for item $a$ between upstream node $j \in L_2$ and downstream node $k \in K$ is computed as follows:

$$\sigma_{a,j,k,t} = \begin{cases} \sqrt{\alpha^2 \sigma_{p,k,t}^2} & if\ CA_{a,j,k,t} = 1 \\ 0 & if\ CA_{a,j,k,t} = 0 \end{cases}, \forall a \in SA, p \in P, j \in L_2, k \in K, t \in T \qquad (3.33)$$

As an illustration, considering that both nodes $k = 5$ and $k = 6$ are supplied by node $j = 3$, the calculation of the demand SD for the subassembly at these nodes is as follows:

$$\sigma_{a,3,5,t} = \sqrt{\alpha^2 \sigma_{p,5}^2} = \sqrt{50^2} = 50, \text{ and}$$

$$\sigma_{a,3,6,t} = \sqrt{\alpha^2 \sigma_{p,6}^2} = \sqrt{75^2} = 75.$$

Likewise, the demand SD for subassembly $a$ at node $j = 4$ is determined as

$$\sigma_{a,4,7,t} = \sqrt{\alpha^2 \sigma_{p,6}^2} = \sqrt{100^2} = 100.$$

Let us suppose that the manufacture of one unit of output item $a$ necessitates the use of $\beta = 2$ units of input item $c$. Based on this relationship, the SD of the demand for item $c$ can be calculated as:

$$\sigma_{c,i,j,t} = \begin{cases} \sqrt{\beta^2 \sigma_{a,j,t}^2} = \sqrt{\beta^2 \sum_{k \in D} \sigma_{a,j,k,t}^2} & if\ CA_{a,j,k,t} = 1 \\ 0 & if\ CA_{a,j,k,t} = 0 \end{cases}, \forall c \in C, a \in SA, p \in P, i \in L_1, j \in L_2, k \in K, t \in T \qquad (3.34)$$

To exemplify, considering that node 1 supplies item $c$ to node 3, which subsequently provides item $a$ to nodes 5 and 6, the SD experienced by node 1 can be calculated as $\sigma_{c,1,3,t} = \sqrt{2^2 \sum_{k\in\{5,6\}} \sigma_{a,3,k}^2} = \sqrt{4(\sigma_{a,3,5}^2 + \sigma_{a,3,6}^2)} = \sqrt{4(50^2 + 75^2)} = 180.28$.

The IP problem discussed in this section is developed based on the foundational model formulated in Section 3.3.1.1, but it is extended to encompass multiple products and multiple periods. Furthermore, our extended model is designed to be adaptable to a broader assembly network, accommodating scenarios where there are multiple final downstream nodes instead of just one, as previously assumed in the original model.

One notable distinction is that our extended model computes the expected SS and the number of shortages based on the connections between two nodes rather than relying on node-level computations.

#### 3.3.2.3 The extended inventory positioning model

**Parameters**

$\mathcal{OC}_{i,j}$: Transit time from component manufacturer $i \in L_1$ to subassembly manufacturer $j \in L_2$;

$\mathcal{OA}_{j,k}$: Transit time from subassembly manufacturer $j \in L_2$ to final assembly manufacturer $k \in K$;

$\mathcal{PC}_i$: Processing time of component manufacturer $i \in L_1$;

$\mathcal{PA}_j$: Processing time of subassembly manufacturer $j \in L_2$;

$\mathcal{PP}_k$: Processing time of final assembly manufacturer $k \in K$;

$\mathcal{HC}_{c,i,t}$: SS holding cost per unit-period of component $c \in C$ from component manufacturer $i \in L_1$ in period $t \in T$;

$\mathcal{HA}_{a,j,t}$: SS holding cost per unit-period of subassembly $a \in SA$ from subassembly manufacturer $j \in L_2$ in period $t \in T$;

$\mathcal{HP}_{p,k,t}$: SS holding cost per unit-period of finished product $p \in P$ from final assembly manufacturer $k \in K$ in period $t \in T$;

$\mathcal{LC}_{c,i,t}$: Shortage cost per unit of component $c \in C$ from component manufacturer $i \in L_1$ in period $t \in T$;

$\mathcal{LA}_{a,j,t}$: Shortage cost per unit of subassembly $a \in SA$ from component manufacturer $j \in L_2$ in period $t \in T$;

$\mathcal{LP}_{p,k,t}$: Shortage cost per unit of finished goods $p \in P$ from final assembly manufacturer $k \in K$ in period $t \in T$;

$\sigma_{p,k,t}$: the SD of demand of finished product $p \in P$ at final assembly manufacturer $k \in K$ in period $t \in T$;

$\sigma_{a,j,k,t}$: the SD of subassembly $a$ at upstream node $j \in L_2$ that is consumed by a downstream node $k \in K$ in period $t \in T$, $\sigma_{a,j,k,t}$ can be computed by Equation (3.33);

$\sigma_{c,i,j,t}$: the SD of component $c$ at upstream node $i \in L_1$ that is consumed by a downstream node $j \in L_2$ in period $t \in T$, $\sigma_{c,i,j,t}$ can be computed by Equation (3.34);

$\mathcal{R}_{k,t}$: Service time committed to the end customer of final assembly manufacturer $k \in K$ in period $t \in T$;

$\mathcal{QC}_{c,i,j,m,t}$: Expected SS of component $c \in C$ held between component manufacturer $i \in L_1$ and subassembly manufacturer $j \in L_2$ in period $t \in T$ if the RT is $m \in M$,

$\mathcal{QA}_{a,j,k,m,t}$: Expected SS of subassembly $a \in SA$ held between subassembly manufacturer $j \in L_2$ final assembly manufacturer $k \in K$ in period $t \in T$ if the RT is $m \in M$;

$\mathcal{QP}_{p,k,m,t}$: Expected SS of finished product $p \in P$ at final assembly manufacturer $k \in K$ in period $t \in T$ if the RT is $m \in M$;

$$\mathcal{QC}_{c,i,j,m,t} = \begin{cases} \Phi^{-1}(s)\sigma_{c,i,j,t}\sqrt{m} & if\ CC_{c,i,j,t} = 1 \\ 0 & if\ CC_{c,i,j,t} = 0 \end{cases}$$
$$\mathcal{QA}_{a,j,k,m,t} = \begin{cases} \Phi^{-1}(s)\sigma_{a,j,k,t}\sqrt{m} & if\ CA_{a,j,k,t} = 1 \\ 0 & if\ CA_{a,j,k,t} = 0 \end{cases} \quad (3.35)$$
$$\mathcal{QP}_{p,k,m,t} = \Phi^{-1}(s)\sigma_{p,k,t}\sqrt{m}$$

$\mathcal{LSC}_{c,i,j,m,t}$: Expected shortage of component $c \in C$ held between component manufacturer $i \in L_1$ and subassembly manufacturer $j \in L_2$ in period $t \in T$ if the RT is $m \in M$,

$\mathcal{LSA}_{a,j,k,m,t}$: Expected shortage of subassembly $a \in SA$ held between subassembly manufacturer $j \in L_2$ final assembly manufacturer $k \in K$ in period $t \in T$ if the RT is $m \in M$;

$\mathcal{LSP}_{p,k,m,t}$: Expected shortage of finished product $p \in P$ at final assembly manufacturer $k \in K$ in period $t \in T$ if the RT is $m \in M$;

$$\mathcal{LSC}_{c,i,j,m,t} = \begin{cases} \mathcal{L}(s)\sigma_{c,i,j,t}\sqrt{m} & if\ CC_{c,i,j,t} = 1 \\ 0 & if\ CC_{c,i,j,t} = 0 \end{cases}$$
$$\mathcal{LSA}_{a,j,k,m,t} = \begin{cases} \mathcal{L}(s)\sigma_{a,j,k,t}\sqrt{m} & if\ CA_{a,j,k,t} = 1 \\ 0 & if\ CA_{a,j,k,t} = 0 \end{cases} \quad (3.36)$$
$$\mathcal{LSP}_{p,k,m,t} = \mathcal{L}(s)\sigma_{p,k,t}\sqrt{m}$$

$\mathcal{M}$: Maximum RT. The possible RTs are listed within a range spanning from zero to a predetermined upper limit. This upper limit correlates with the total time of the longest path within the network. Specifically, it encompasses the time it takes to traverse from one of the most upstream nodes to one of the final nodes. The longest path is determined by the longest path model, as shown in Appendix B.

**Decision variables**

$\mathcal{SC}^{in}_{c,i,j,t}$: Incoming service time committing to deliver component $c \in C$ from component manufacturer $i \in L_1$ to subassembly manufacturer $j \in L_2$ in period $t \in T$;

$\mathcal{SA}^{in}_{a,j,k,t}$: Incoming service time committing to deliver subassembly $a \in SA$ from subassembly manufacturer $j \in L_2$ to final assembly manufacturer $k \in K$ in period $t \in T$;

$\mathcal{SC}^{out}_{c,i,j,t}$: Outgoing service time committing to deliver component $c \in C$ from component manufacturer $i \in L_1$ to subassembly manufacturer $j \in L_2$ in period $t \in T$;

$\mathcal{SA}^{out}_{a,j,k,t}$: Outgoing service time committing to deliver subassembly $a \in SA$ from subassembly manufacturer $j \in L_2$ to final assembly manufacturer $k \in K$ in period $t \in T$;

$\mathcal{SP}^{out}_{p,k,t}$: Outgoing service time to deliver finished product $p \in P$ of final assembly manufacturer $k \in K$ in period $t \in T$;

$\mathcal{XC}_{c,i,j,t}$: RT of output component $c \in C$ at component manufacturer $i \in L_1$, indicating a need for SS if positive, serving subassembly manufacturer $j \in L_2$ in period $t \in T$;

$\mathcal{XA}_{a,j,k,t}$: RT of output subassembly $a \in SA$ at subassembly manufacturer $j \in L_2$, indicating a need for SS if positive, serving final assembly manufacturer $k \in K$ in period $t \in T$;

$\mathcal{XP}_{p,k,t}$: RT of finished product $p \in P$ at final assembly manufacturer $k \in K$ in period $t \in T$;

$\mathcal{UC}_{c,i,j,m,t}$: Binary variable, 1 if $\mathcal{XC}_{c,i,j,t} = m$ in period $t \in T$, or 0 otherwise;

$\mathcal{UA}_{a,j,k,m,t}$: Binary variable, 1 if $\mathcal{XA}_{a,j,k,t} = m$ in period $t \in T$, or 0 otherwise;

$\mathcal{UP}_{p,k,m,t}$: Binary variable, 1 if $\mathcal{XP}_{p,k,t} = m$ in period $t \in T$, or 0 otherwise;

$\mathcal{YC}_{c,i,j,t}$: RT of subassembly manufacturer $j \in L_2$ which requires component $c \in C$ from component manufacturer $i \in L_1$ as input item in period $t \in T$;

$\mathcal{YA}_{a,j,k,t}$: RT of final assembly manufacturer $k \in K$ which requires subassembly $a \in SA$ from subassembly manufacturer $j \in L_2$ as input item in period $t \in T$;

$\mathcal{VC}_{c,i,j,m,t}$: Binary variable, 1 if $\mathcal{YC}_{c,i,j,t} = m$ in period $t \in T$, or 0 otherwise;

$\mathcal{VA}_{a,j,k,m,t}$: Binary variable, 1 if $\mathcal{YA}_{a,j,k,t} = m$ in period $t \in T$, or 0 otherwise;

**The MILP model:**

The aim of the IP problem is to minimize the overall cost associated with SS, including both the holding cost and the cost caused by shortages of both output and input items. the objective function is expressed as follows:

$$
\begin{aligned}
Minimize\ Z_2 = & \sum_{c\in C}\sum_{i\in L_1}\sum_{j\in L_2}\sum_{m\in M}\sum_{t\in T} \mathcal{HC}_{c,i,t}\mathcal{QC}_{c,i,j,m,t}\,(\mathcal{UC}_{c,i,j,m,t} + \mathcal{VC}_{c,i,j,m,t}) \\
& + \sum_{a\in A}\sum_{j\in L_2}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T} \mathcal{HA}_{a,j,t}\mathcal{QA}_{a,j,k,m,t}(\mathcal{UA}_{a,j,k,m,t} + \mathcal{VA}_{c,i,j,m,t}) \\
& + \sum_{p\in P}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T} \mathcal{HP}_{p,k,t}\mathcal{QP}_{p,k,m}\,\mathcal{UP}_{p,k,m,t} \\
& + \sum_{c\in C}\sum_{i\in L_1}\sum_{j\in L_2}\sum_{m\in M}\sum_{t\in T} \frac{365}{m} \times \mathcal{LC}_{c,i,t}\mathcal{LSC}_{c,i,j,m,t}\,(\mathcal{UC}_{c,i,j,m,t} + \mathcal{VC}_{c,i,j,m,t}) \\
& + \sum_{a\in A}\sum_{j\in L_2}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T} \frac{365}{m} \times \mathcal{LA}_{a,j,t}\mathcal{LSA}_{a,j,k,m,t}(\mathcal{UA}_{a,j,k,m,t} + \mathcal{VA}_{a,j,k,m,t})
\end{aligned}
\tag{3.37}
$$

$$+\sum_{p\in P}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T}\frac{365}{m}\times \mathcal{LP}_{p,k,t}\mathcal{LSP}_{p,k,m}\,\mathcal{UP}_{p,k,m,t}$$

Subject to

$$\mathcal{SC}^{out}_{c,i,j,t}-\mathcal{SC}^{in}_{c,i,j,t}+\mathcal{OC}_{i,j}\le \mathcal{YC}_{c,i,j,t}\quad \forall c\in C, i\in L_1,\ j\in L_2, t\in T \tag{3.38}$$

$$\mathcal{SA}^{out}_{a,j,k,t}-\mathcal{SA}^{in}_{a,j,k,t}+\mathcal{OA}_{j,k}\le \mathcal{YA}_{a,j,k,t}\quad \forall a\in SA, j\in L_2,\ k\in K, t\in T \tag{3.39}$$

$$-\mathcal{SC}^{out}_{c,i,j,t}+\mathcal{PC}_i\le \mathcal{XC}_{c,i,j,t}\qquad \forall c\in C, i\in L_1, j\in L_2, t\in T \tag{3.40}$$

$$\mathcal{SC}^{in}_{c,i,j,t}-\mathcal{SA}^{out}_{a,j,k,t}+\mathcal{PA}_j\le \mathcal{XA}_{a,j,k,t}\qquad \forall (c,a)\in BOMC, i\in L_1, j\in L_2, k\in K, t\in T \tag{3.41}$$

$$\mathcal{SA}^{in}_{a,j,k,t}-\mathcal{SP}^{out}_{p,k,t}+\mathcal{PP}_k\le \mathcal{XP}_{p,k,t}\qquad \forall (a,p)\in BOMA, j\in L_2, k\in K, t\in T \tag{3.42}$$

$$\mathcal{SC}^{in}_{c,i,j,t}\le \mathcal{M}\qquad \forall c\in C,\ i\in L_1, j\in L_2, t\in T \tag{3.43}$$

$$\mathcal{SA}^{in}_{a,j,k,t}\le \mathcal{M}\qquad \forall a\in SA,\ j\in L_2, k\in K, t\in T \tag{3.44}$$

$$\mathcal{SC}^{out}_{c,i,j,t}\le \mathcal{M}\qquad \forall c\in C,\ i\in L_1, j\in L_2, t\in T \tag{3.45}$$

$$\mathcal{SA}^{out}_{a,j,k,t}\le \mathcal{M}\qquad \forall a\in SA,\ j\in L_2, k\in K, t\in T \tag{3.46}$$

$$\mathcal{SP}^{out}_{p,k,t}\le \mathcal{R}_{k,t}\quad \forall p\in P, k\in K, t\in T \tag{3.47}$$

$$\mathcal{XC}_{c,i,j,t}=\sum_{m\in M} m\,\mathcal{UC}_{c,i,j,m,t}\qquad \forall c\in C,\ i\in L_1, j\in L_2, t\in T \tag{3.48}$$

$$\mathcal{XA}_{a,j,k,t}=\sum_{m\in M} m\,\mathcal{UA}_{a,j,k,m,t}\qquad \forall a\in SA,\ j\in L_2, k\in K, t\in T \tag{3.49}$$

$$\mathcal{XP}_{p,k,t}=\sum_{m\in M} m\,\mathcal{UP}_{p,k,m,t}\qquad \forall p\in P,\ k\in K, t\in T \tag{3.50}$$

$$\mathcal{YC}_{c,i,j,t}=\sum_{m\in M} m\,\mathcal{VC}_{c,i,j,m,t}\qquad \forall c\in C, i\in L_1, j\in L_2, t\in T \tag{3.51}$$

$$\mathcal{YA}_{a,j,k,t}=\sum_{m\in M} m\,\mathcal{VA}_{a,j,k,m,t}\qquad \forall a\in SA, j\in L_2, k\in K, t\in T \tag{3.52}$$

$$\sum_{m\in M}\mathcal{UC}_{c,i,j,m,t}\le 1\qquad \forall c\in C,\ i\in L_1, j\in L_2, t\in T \tag{3.53}$$

$$\sum_{m\in M}\mathcal{UA}_{a,j,k,m,t}\le 1\qquad \forall a\in SA,\ j\in L_2, k\in K, t\in T \tag{3.54}$$

$$\sum_{m \in M} \mathcal{UP}_{p,k,m,t} \leq 1 \qquad \forall p \in P,\ k \in K, t \in T \tag{3.55}$$

$$\sum_{m \in M} \mathcal{VC}_{c,i,j,m,t} \leq 1 \qquad \forall c \in C, i \in L_1, j \in L_2, t \in T \tag{3.56}$$

$$\sum_{m \in M} \mathcal{VA}_{a,j,k,m,t} \leq 1 \qquad \forall a \in SA,\ j \in L_2, k \in K, t \in T \tag{3.57}$$

$$\begin{aligned} &\mathcal{SC}^{in}_{c,i,j,t}, \mathcal{SA}^{in}_{a,j,k,t}, \mathcal{SC}^{out}_{c,i,j,t}, \mathcal{SA}^{out}_{a,j,k,t}, \mathcal{SP}^{out}_{p,k,t}, \mathcal{XC}_{c,i,j,t}, \mathcal{XA}_{a,j,k,t}, \mathcal{XP}_{p,k,t}, \mathcal{YC}_{c,i,j,t}, \\ &\mathcal{YA}_{a,j,k,t} \geq 0, \quad \forall c \in C, a \in SA, p \in P, i \in L_1, j \in L_2, k \in K, m \in M, t \in T \end{aligned} \tag{3.58}$$

$$\begin{aligned} &\mathcal{UC}_{c,i,j,m,t}, \mathcal{UA}_{a,j,k,m,t}, \mathcal{UP}_{p,k,m,t}, \mathcal{VC}_{c,i,j,m,t}, \mathcal{VA}_{a,j,k,m,t} \in \{0,1\} \\ &\forall c \in C, a \in SA, p \in P, i \in L_1 \cup\ L_2, j \in L_2 \cup K, m \in M, t \in T \end{aligned} \tag{3.59}$$

Constraints (3.38) and (3.39) are to define the RT for different input items, whereas Constraints (3.40) to (3.41) are dedicated to output items. These constraints are derived from Equation (3.32) to capture the timing relationships within the system. In addition, Constraints (3.43) - (3.46) are to enforce that the incoming and outgoing service times must not exceed the maximum RT. To prevent the violation of service commitments, Constraints (3.47) enforce that the outgoing service times at the final nodes remain within the bounds of the service times committed to the customers. Constraints (3.48) - (3.52) are responsible for identifying the values of the RTs times by the selected option of the RT for both output and input items. Constraints (3.53) - (3.57) ensure that each arc, during each period, is aligned with a single selected option for the RT. Lastly, Constraints (3.58) and (3.59) are binary and non-negativity constraints.

### 3.4 The integrated model

#### 3.4.1 An alternative approach for SS computation

When integrating the NDP and IP, it is essential to note that the network structure is initially unknown. Additionally, the SD along an arc connecting two nodes depends on whether a flow traverses that arc. Consequently, the approach outlined in the previous section is not suitable for computing the SD in this context. This section presents an alternative approach to deal with this issue.

Specifically, given a set of product flows within the network, the variance associated with each arc can be computed using proportions derived from the total demand variance through that specific arc.

In the integrated model, the determination of flows and the specification of variance on the arcs are intertwined. This interdependence complicates the computation of SDs. To tackle this complexity, we rely on a set of binary variables that cover all combinations of demand variance between two layers. These combinations serve to define the chosen proportions, which in turn specify the calculation of SD values.

Consider the case of single period, within layer $L_2$, it is demonstrated that the SD of an item $a$ along the arc connecting node $j \in L_2$ to node $k \in K$ can be calculated as $\sigma_{a,j,k} = \sqrt{\alpha^2 \sigma_{p,k}^2}, \forall a \in SA, p \in P, j \in L_2, k \in K$.

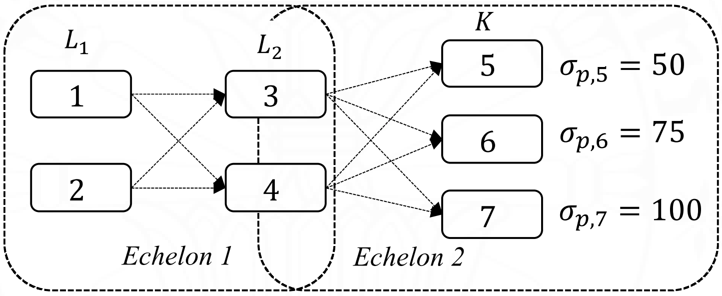



**Figure 3.2** Simple supply chain with all possible connections

Considering the network illustrated in Figure 3.2, there are a total of seven possible combinations of demand variance proportions. These combinations are interpreted through the parameters $\kappa_{p,1}$, $\kappa_{p,2}, \kappa_{p,3}$, representing the demand variance proportions for product $p$ in scenarios where either node 3 or node 4 serves each of the demand nodes 5, 6, and 7, respectively. Furthermore, $\kappa_{p,4}, \kappa_{p,5}$, and $\kappa_{p,6}$ denote the demand variance proportions in instances where either node 3 or node 4 serves two out of the three nodes, namely nodes 5 and 6, nodes 5 and 7, and nodes 6 and 7. Additionally, variable $\kappa_{p,7}$ corresponds to situations where either node 3 or node 4 serves all three demand nodes.

The computation of these variance proportions is outlined as follows:

$$\kappa_{p,1} = \frac{\sigma_{p,5}^2}{\sum_{k \in D} \sigma_{p,k}^2} = \frac{50^2}{50^2 + 75^2 + 100^2} = 0.1379$$

$$\kappa_{p,2} = \frac{\sigma_{p,6}^2}{\sum_{k\in D}\sigma_{p,k}^2} = \frac{75^2}{50^2+75^2+100^2} = 0.3103$$

$$\kappa_{p,3} = \frac{\sigma_{p,7}^2}{\sum_{k\in D}\sigma_{p,k}^2} = \frac{100^2}{50^2+75^2+100^2} = 0.5517$$

For layer $L_2 = \{3,4\}$ serving layer $K = \{5,6,7\}$, the SDs of $a$ on the arc from node $j \in L_2$ to node $k \in K$, $\sigma_{a,j,k}$, are computed as

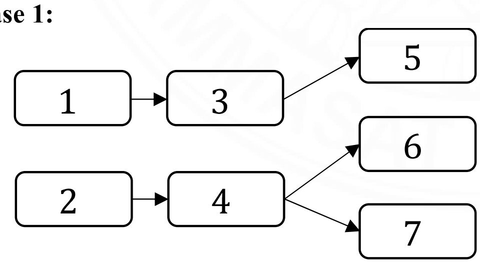
$$\sigma_{a,j,5} = \sqrt{\alpha^2\sigma_{p,5}^2} = \sqrt{\kappa_{p,1}\alpha^2\sum_{k\in D}\sigma_{p,k}^2} = 50,$$

$$\sigma_{a,j,6} = \sqrt{\alpha^2\sigma_{p,6}^2} = \sqrt{\kappa_{p,2}\alpha^2\sum_{k\in D}\sigma_{p,k}^2} = 75, \text{and}$$

$$\sigma_{a,j,7} = \sqrt{\alpha^2\sigma_{p,7}^2} = \sqrt{\kappa_{p,3}\alpha^2\sum_{k\in D}\sigma_{p,k}^2} = 100.$$

Moving to the upstream layer denoted as $L_1 = \{1,2\}$, let's consider a scenario where we require $\beta = 2$ units of input item $c$ to manufacture 1 unit of output item $a$ at downstream node $j \in L_2 = \{3,4\}$. In this context, the SD of $c$ along the arc from $i$ to $j$, $\sigma_{c,i,j}$, can be expressed in eight distinct cases as detailed below. It should be noted that these cases assume that node 1 supplies to node 3 and node 2 supplies to node 4.

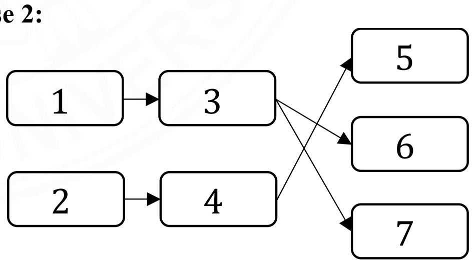
**Case 1:**




$$\sigma_{c,1,3} = \sqrt{\beta^2\alpha^2\sigma_{a,3}^2} = \sqrt{\beta^2\sigma_{a,3,5}^2}$$
$$= \sqrt{\beta^2\kappa_{p,1}\sum_{k\in D}\sigma_{p,k}^2} = 100$$

$$\sigma_{c,2,4} = \sqrt{\beta^2\alpha^2\sigma_{a,4}^2} = \sqrt{\beta^2(\sigma_{a,4,6}^2+\sigma_{a,4,7}^2)}$$
$$= \sqrt{\beta^2(\kappa_{p,2}+\kappa_{p,3})\sum_{k\in D}\sigma_{p,k}^2}$$
$$= 250$$

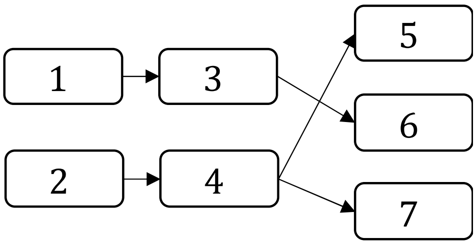
**Case 2:**




$$\sigma_{c,1,3} = \sqrt{\beta^2\alpha^2\sigma_{a,3}^2} = \sqrt{\beta^2(\sigma_{a,3,6}^2+\sigma_{a,3,7}^2)}$$
$$= \sqrt{\beta^2(\kappa_{p,2}+\kappa_{p,3})\sum_{k\in D}\sigma_{p,k}^2}$$
$$= 250$$

$$\sigma_{c,2,4} = \sqrt{\beta^2\alpha^2\sigma_{a,4}^2} = \sqrt{\sigma_{a,4,5}^2} = \sqrt{\kappa_{p,1}\sum_{k\in D}\sigma_{p,k}^2}$$
$$= 100$$

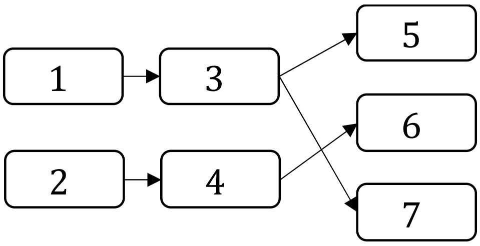
**Case 3:**




$$\sigma_{c,1,3} = \sqrt{\beta^2\alpha^2\sigma_{a,3}^2} = \sqrt{\beta^2\sigma_{a,3,6}^2} = \sqrt{\beta^2\kappa_{p,2}\sum_{k\in D}\sigma_{p,k}^2} = 150$$

$$\sigma_{c,2,4} = \sqrt{\beta^2\alpha^2\sigma_{a,4}^2} = \sqrt{\beta^2(\sigma_{a,4,5}^2+\sigma_{a,4,7}^2)} = \sqrt{\beta^2(\kappa_{p,1}+\kappa_{p,3})\sum_{k\in D}\sigma_{p,k}^2} = 223.61$$

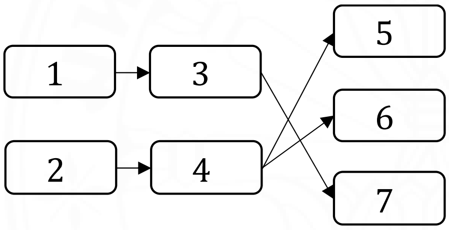
**Case 4:**




$$\sigma_{c,1,3} = \sqrt{\beta^2\alpha^2\sigma_{a,3}^2} = \sqrt{\beta^2(\sigma_{a,3,5}^2+\sigma_{a,3,7}^2)} = \sqrt{\beta^2(\kappa_{p,1}+\kappa_{p,3})\sum_{k\in D}\sigma_{p,k}^2} = 223.61$$

$$\sigma_{c,2,4} = \sqrt{\beta^2\alpha^2\sigma_{a,4}^2} = \sqrt{\beta^2\sigma_{a,4,6}^2} = \sqrt{\beta^2\kappa_{p,2}\sum_{k\in D}\sigma_{p,k}^2} = 150$$

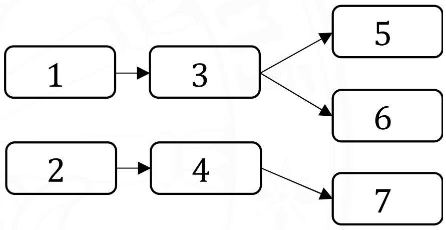
**Case 5:**




$$\sigma_{c,1,3} = \sqrt{\beta^2\alpha^2\sigma_{a,3}^2} = \sqrt{\beta^2\sigma_{a,3,7}^2} = \sqrt{\beta^2\kappa_{p,3}\sum_{k\in D}\sigma_{p,k}^2} = 200$$

$$\sigma_{c,2,4} = \sqrt{\beta^2\alpha^2\sigma_{a,4}^2} = \sqrt{\beta^2(\sigma_{a,4,5}^2+\sigma_{a,4,6}^2)} = \sqrt{\beta^2(\kappa_{p,1}+\kappa_{p,2})\sum_{k\in D}\sigma_{p,k}^2} = 180.28$$

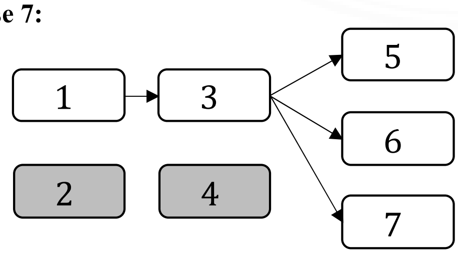
**Case 6:**




$$\sigma_{c,1,3} = \sqrt{\beta^2\alpha^2\sigma_{a,3}^2} = \sqrt{\beta^2(\sigma_{a,3,5}^2+\sigma_{a,3,6}^2)} = \sqrt{\beta^2(\kappa_{p,1}+\kappa_{p,2})\sum_{k\in D}\sigma_{p,k}^2} = 180.28$$

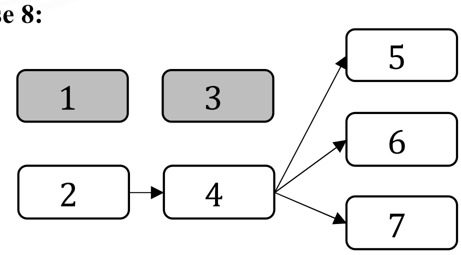
$$\sigma_{c,2,4} = \sqrt{\beta^2\alpha^2\sigma_{a,4}^2} = \sqrt{\beta^2\sigma_{a,4,7}^2} = \sqrt{\beta^2\kappa_{p,3}\sum_{k\in D}\sigma_{p,k}^2} = 200$$

**Case 7:**




$$\sigma_{c,1,3} = \sqrt{\beta^2\alpha^2\sigma_{a,3}^2} = \sqrt{\beta^2(\sigma_{a,3,5}^2+\sigma_{a,3,6}^2+\sigma_{a,3,7}^2)} = \sqrt{\beta^2(\kappa_{p,1}+\kappa_{p,2}+\kappa_{p,3})\sum_{k\in D}\sigma_{p,k}^2} = 269.26$$

**Case 8:**




$$\sigma_{c,2,4} = \sqrt{\beta^2\alpha^2\sigma_{a,4}^2} = \sqrt{\beta^2(\sigma_{a,4,5}^2+\sigma_{a,4,6}^2+\sigma_{a,4,7}^2)} = \sqrt{\beta^2(\kappa_{p,1}+\kappa_{p,2}+\kappa_{p,3})\sum_{k\in D}\sigma_{p,k}^2} = 269.26$$

**Figure 3.3** Eight cases of SD computation

When the configuration is such that node 1 supplies to node 4 and node 2 supplies to node 3, a similar set of eight cases is generated, as detailed above.

For a given network, we introduce the notation $\mathcal{ED}_{p,o}$, which represents the multiplier $o$ utilized for calculating the SD of demand when an upstream node serves downstream nodes producing the finished product $p$. $\mathcal{ED}_{p,o}$ is expressed as follows.

**Table 3.1** Multipliers of layer $K$

| $o$ | $K^*$ | $\mathcal{ED}_{p,o}$ |
|---|---|---|
| 1 | {5} | $\sqrt{\kappa_{p,1}}$ |
| 2 | {6} | $\sqrt{\kappa_{p,2}}$ |
| 3 | {7} | $\sqrt{\kappa_{p,3}}$ |

The notation $K^*$ signifies the collection of all downstream nodes that are supplied by a particular upstream node. For instance, $K^* = \{5\}$ indicates that an upstream node exclusively provides sub-assembly to node 5.

Similarly, a multiplier $\mathcal{EA}_{a,o}$ is introduced for subassembly $a$ from layer $L_2$, where $(a, p) \in BOMA$. The expression for $\mathcal{EA}_{a,o}$ is as follows:

**Table 3.2** Multipliers of layer $L_2$

| $o$ | $K^*$ | $\mathcal{EA}_{a,o}$ |
|---|---|---|
| 1 | {5} | $\sqrt{\kappa_{p,1}}$ |
| 2 | {6} | $\sqrt{\kappa_{p,2}}$ |
| 3 | {7} | $\sqrt{\kappa_{p,3}}$ |
| 4 | {5, 6} | $\sqrt{\kappa_{p,1} + \kappa_{p,2}}$ |
| 5 | {5, 7} | $\sqrt{\kappa_{p,1} + \kappa_{p,3}}$ |
| 6 | {6, 7} | $\sqrt{\kappa_{p,2} + \kappa_{p,3}}$ |
| 7 | {5, 6, 7} | $\sqrt{\kappa_{p,1} + \kappa_{p,2} + \kappa_{p,3}}$ |

The number of multipliers expands with the augmentation of nodes within the final downstream layer. Specifically, if the final layer encompasses $|K|$ nodes, then the potential number of multipliers can be determined using the expression $\sum_{r=1}^{|K|} C(|K|, r) = \sum_{r=1}^{|K|} \frac{|K|!}{(|K|-r)!r!} = (2^{|K|} - 1)$, as dictated by the binomial theorem.

Extending this approach to multiple periods, let us denote two binary variables as $\mathcal{ZA}_{a,j,k,o,t}$ and $\mathcal{ZC}_{c,i,j,o,t}$. These binary variables signify the choice of multiplier $o \in OPTA$ at node $j \in L_2$ serving node $k \in K$ with component $a$, and the choice of multiplier $o \in OPTC$ at node $i \in L_1$ serving node $j \in L_2$ with component $c$, respectively.

The equations to compute SDs are derived as follows:

$$\sigma_{a,j,k,t} = \mathcal{ZA}_{a,j,k,o,t}\mathcal{ED}_{p,o}\sqrt{\alpha^2\sum_{k\in K}\sigma_{p,k,t}^2}, \quad \forall (a,p) \in BOMA\, j \in L_2, k \in K, o \in OPTA, t \in T \tag{3.60}$$

$$\sigma_{c,i,j,t} = \mathcal{ZC}_{c,i,j,o,t}\mathcal{EA}_{a,o}\sqrt{\beta^2\alpha^2\sum_{k\in K}\sigma_{p,k,t}^2}, \quad \forall (c,a) \in BOMC, (a,p) \in BOMA, i \in L_1, j \in L_2, o \in OPTC, t \in T \tag{3.61}$$

When assuming a normal distribution for demand, the SS quantity for an item, whether it originates as an output from node $i$ or as an input from node $j$, is typically defined in the following manner:

$$\begin{aligned}\mathcal{KA}_{a,j,k,m,t}\ (or\ \mathcal{GA}_{a,j,k,m,t}) &= \Phi^{-1}(s)\sigma_{a,j,k,t}\sqrt{m} \\ &= \Phi^{-1}(s)\mathcal{ZA}_{a,j,k,o,t}\mathcal{ED}_{p,o}\sqrt{\alpha^2\sum_{k\in K}\sigma_{p,k,t}^2}\sqrt{m},\end{aligned} \quad \forall (a,p) \in BOMA, j \in L_2, k \in K, o \in OPTA, m \in M, t \in T \tag{3.62}$$

$$\begin{aligned}\mathcal{KC}_{c,i,j,m,t}\ (or\ \mathcal{GC}_{c,i,j,m,t}) &= \Phi^{-1}(s)\sigma_{c,i,j,t}\sqrt{m} \\ &= \Phi^{-1}(s)\mathcal{ZC}_{c,i,j,o,t}\mathcal{EA}_{a,o}\sqrt{\beta^2\alpha^2\sum_{k\in K}\sigma_{p,k,t}^2}\sqrt{m},\end{aligned} \quad \forall (c,a) \in BOMC, (a,p) \in BOMA, i \in L_1, j \in L_2, o \in OPTC, m \in M, t \in T \tag{3.63}$$

Let $\mathcal{QA}_{a,j,m} = \Phi^{-1}(s)\sqrt{\alpha^2\sum_{k\in K}\sigma_{p,k,t}^2}\sqrt{m}$, and $\mathcal{QC}_{c,i,m} = \Phi^{-1}(s)\sqrt{\beta^2\alpha^2\sum_{k\in K}\sigma_{p,k,t}^2}\sqrt{m}$, the SS amount of an item are re-written as,

$$\mathcal{KA}_{a,j,k,m,t}\ \left(or\ \mathcal{GA}_{a,j,k,m,t}\right) = \mathcal{ZA}_{a,j,k,o,t}\mathcal{ED}_{p,o}\mathcal{QA}_{a,j,m,t},$$
$$\forall (a,p) \in BOMA, j \in L_2, k \in K, o \in OPTA, m \in M, t \in T \quad (3.64)$$

$$\mathcal{KC}_{c,i,j,m,t}\ \left(or\ \mathcal{GC}_{c,i,j,m,t}\right) = \mathcal{ZC}_{c,i,j,o,t}\mathcal{EA}_{a,o}\mathcal{QC}_{c,i,m,t},$$
$$\forall (c,a) \in BOMC, (a,p) \in BOMA, i \in L_1, j \in L_2, o \in OPTC, m \in M, t \in T \quad (3.65)$$

In general, the unfulfilled fraction of the demand is assumed to be backlogged. The backlog expected value is computed through the following equations:

$$\mathcal{JC}_{c,i,j,m,t}\ (or\ \mathcal{WC}_{c,i,j,m,t}) = \mathcal{L}(s)\sigma_{c,i,j,t}\sqrt{m}$$

$$\mathcal{JA}_{a,j,k,m,t}\ (or\ \mathcal{WA}_{a,j,k,m,t}) = \mathcal{L}(s)\sigma_{a,j,k,t}\sqrt{m}$$

Similarly, let $\mathcal{LSA}_{a,j,m,t} = \mathcal{L}(s)\sqrt{\alpha^2 \sum_{k\in K} \sigma_{p,k,t}^2}\,\sqrt{m}$, and $\mathcal{LSC}_{c,i,m,t} = \mathcal{L}(s)\sqrt{\beta^2\alpha^2 \sum_{k\in K} \sigma_{p,k,t}^2}\,\sqrt{m}$, we have,

$$\mathcal{JA}_{a,j,k,m,t}\ (or\ \mathcal{WA}_{a,j,k,m,t}) = \mathcal{ZA}_{a,j,k,o,t}\mathcal{ED}_{p,o}\mathcal{LSA}_{a,j,m,t},$$
$$\forall (a,p) \in BOM_A, j \in L_2, k \in K, o \in OPT_A, m \in M, t \in T, \text{and} \quad (3.66)$$

$$\mathcal{JC}_{c,i,j,m,t}\ \left(or\ \mathcal{WC}_{c,i,j,m,t}\right) = \mathcal{ZC}_{c,i,j,o,t}\mathcal{EA}_{a,o}\mathcal{LSC}_{c,i,m,t},$$
$$\forall (c,a) \in BOM_C, (a,p) \in BOM_A, i \in L_1, j \in L_2, o \in OPT_C, m \in M, t \in T \quad (3.67)$$

### 3.4.2 The MILP of the integrated model:

The two models can be integrated by using additional parameters and decision variables. It should be noted that the proposed preprocessing approach in the previous section is for linearizing the integrated model. The original mixed-integer nonlinear programming model is presented in Appendix A.

**Parameters:**

$\mathcal{A}_{a,j,k,o}$: 1 if there exists an arc subassembly $a \in SA$ at subassembly manufacturer $j \in L_2$ is served to final assembly manufacturer $k \in K$ with option $o \in OPT$, or 0 otherwise;

With SS and shortage expressions as presented above, the objective function of the IP model is described as follows:

$$
\begin{aligned}
Z_3 = &\sum_{c\in C}\sum_{i\in L_1}\sum_{j\in L_2}\sum_{m\in M}\sum_{t\in T}\mathcal{HC}_{c,i,t}\left(\mathcal{KC}_{c,i,j,m,t}+\mathcal{GC}_{c,i,j,m,t}\right)\\
&+\sum_{a\in A}\sum_{j\in L_2}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T}\mathcal{HA}_{a,j,t}\left(\mathcal{KA}_{a,j,k,m,t}+\mathcal{GA}_{a,j,k,m,t}\right)\\
&+\sum_{p\in P}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T}\mathcal{HP}_{p,k,t}\mathcal{KP}_{p,k,m,t}\\
&+\sum_{c\in C}\sum_{i\in L_1}\sum_{j\in L_2}\sum_{m\in M}\sum_{t\in T}\frac{365}{m}\times\mathcal{LC}_{c,i,t}\left(\mathcal{JC}_{c,i,j,m,t}+\mathcal{WC}_{c,i,j,m,t}\right)\\
&+\sum_{a\in A}\sum_{j\in L_2}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T}\frac{365}{m}\times\mathcal{LA}_{a,j,t}\left(\mathcal{JA}_{a,j,k,m,t}+\mathcal{WA}_{a,j,k,m,t}\right)\\
&+\sum_{p\in P}\sum_{k\in K}\sum_{m\in M}\sum_{t\in T}\frac{365}{m}\times\mathcal{LP}_{p,k,t}\mathcal{JP}_{p,k,m,t}
\end{aligned}
\tag{3.68}
$$

From the modification of the IP problem, the integrated model is expressed below.

$$Minimize\ TAC = Z_1 + Z_3$$

Subject to all the constraints from both NDP and IP models. Additionally, some constraints are added for the linearization purpose.

$$\mathcal{SC}^{out}_{c,i,j,t} - \mathcal{SC}^{in}_{c,i,j,t} + \mathcal{OC}_{i,j}CC_{c,i,j,t} \le \mathcal{YC}_{c,i,j,t} \quad \forall c\in C, i\in L_1, j\in L_2, t\in T \tag{3.69}$$

$$\mathcal{SA}^{out}_{a,j,k,t} - \mathcal{SA}^{in}_{a,j,k,t} + \mathcal{OA}_{j,k}CA_{a,j,k,t} \le \mathcal{YA}_{a,j,k,t} \quad \forall a\in SA, j\in L_2, k\in D, t\in T \tag{3.70}$$

$$-\mathcal{SC}^{out}_{c,i,j,t} + \mathcal{PC}_i - \left(1-CC_{c,i,j,t}\right)\mathcal{M} \le \mathcal{XC}_{c,i,j,t} \quad \forall c\in C, i\in L_1, j\in L_2, t\in T \tag{3.71}$$

$$\begin{aligned}\mathcal{SC}^{in}_{c,i,j,t} - \mathcal{SA}^{out}_{a,j,k,t} + \mathcal{PA}_j - \left(1-CC_{c,i,j,t}\right)\mathcal{M} - \left(1-CA_{a,j,k,t}\right)\mathcal{M}\\ \le \mathcal{XA}_{a,j,k,t}\\ \forall (c,a)\in BOMC, i\in L_1, j\in L_2, k\in D, t\in T\end{aligned} \tag{3.72}$$

$$\mathcal{SA}^{in}_{a,j,k,t} - \mathcal{SP}^{out}_{p,k,t} + \mathcal{PP}_k - \left(1-CA_{a,j,k,t}\right)\mathcal{M} \le \mathcal{XP}_{p,k,t} \quad \forall (a,p)\in BOMA, j\in L_2, k\in D, t\in T \tag{3.73}$$

$$\sum_{m\in M}\mathcal{UC}_{c,i,j,m,t} \le ZC_{i,t} \qquad \forall c\in C, i\in L_1, j\in L_2, t\in T \tag{3.74}$$

$$\sum_{m\in M}\mathcal{UA}_{a,j,k,m,t} \le ZA_{j,t} \qquad \forall a\in SA, j\in L_2, k\in K, t\in T \tag{3.75}$$

$$\sum_{m\in M}\mathcal{VC}_{c,i,j,m,t} \le CC_{c,i,j,t} \qquad \forall c\in C, i\in L_1, j\in L_2, t\in T \tag{3.76}$$

$$\sum_{m\in M}\mathcal{VA}_{a,j,k,m,t} \le CA_{a,j,k,t} \qquad \forall a\in SA, j\in L_2, k\in K, t\in T \tag{3.77}$$

$$\sum_{o\in OPT_C}\sum_{k\in K}\mathcal{A}_{a,j,k,o}\mathcal{ZC}_{c,i,j,o,t} \geq \sum_{k\in K} CA_{a,j,k,t} - \left(1 - CC_{c,i,j,t}\right)BIG \quad \forall (c,a) \in BOMC, i \in L_1, j \in L_2, t \in T \tag{3.78}$$

$$\sum_{o\in OPT_C}\mathcal{A}_{a,j,k,o}\mathcal{ZC}_{c,i,j,o,t} \leq CA_{a,j,k,t} \quad \forall (c,a) \in BOMC, i \in L_1, j \in L_2, k \in K, t \in T \tag{3.79}$$

$$\sum_{o\in OPT_A}\mathcal{A}_{a,j,k,o}\mathcal{ZA}_{a,j,k,o,t} \geq CA_{a,j,k,t} \quad \forall a \in SA, j \in L_2, k \in K, t \in T \tag{3.80}$$

$$\sum_{k\in D}\sum_{o\in OPT_A}\mathcal{A}_{a,j,k,o}\mathcal{ZA}_{a,j,k,o,t} \leq 1 \quad \forall a \in SA, j \in L_2, t \in T \tag{3.81}$$

$$\sum_{o\in OPT_C}\mathcal{ZC}_{c,i,j,o,t} \leq CC_{c,i,j,t} \quad \forall c \in C, i \in L_1, j \in L_2, t \in T \tag{3.82}$$

$$\sum_{o\in OPT_A}\mathcal{ZA}_{a,j,k,o,t} \leq CA_{a,j,k,t} \quad \forall a \in SA, j \in L_2, k \in K, t \in T \tag{3.83}$$

$$\sum_{o\in OPT_C}\mathcal{ZC}_{c,i,j,o,t} \geq \sum_{m\in M}\mathcal{UC}_{c,i,j,m,t} \quad \forall \mathrm{c} \in C, i \in L_1, j \in L_2, t \in T \tag{3.84}$$

$$\sum_{o\in OPT_A}\mathcal{ZA}_{a,j,k,o,t} \geq \sum_{m\in M}\mathcal{UA}_{a,j,k,m,t} \quad \forall a \in SA, j \in L_2, k \in K, t \in T \tag{3.85}$$

$$\sum_{o\in OPT_C}\mathcal{ZC}_{c,i,j,o,t} \geq \sum_{m\in M}\mathcal{VC}_{c,i,j,m,t} \quad \forall c \in C, i \in L_1, j \in L_2, t \in T \tag{3.86}$$

$$\sum_{o\in OPT_A}\mathcal{ZA}_{a,j,k,o,t} \geq \sum_{m\in M}\mathcal{VA}_{a,j,k,m,t} \quad \forall a \in SA, j \in L_2, k \in K, t \in T \tag{3.87}$$

$$\mathcal{KC}_{c,i,j,m,t} \geq \sum_{o\in OPT_C}\mathcal{QC}_{c,i,m}\mathcal{EA}_{a,o}\mathcal{ZC}_{c,i,j,o,t} - \left(1 - \mathcal{UC}_{c,i,j,m,t}\right)BIG \quad \forall (c,a) \in BOMC, i \in L_1, j \in L_2, m \in M, t \in T \tag{3.88}$$

$$\mathcal{KA}_{a,j,k,m,t} \geq \sum_{o\in OPT_A}\mathcal{QA}_{a,j,m}\mathcal{ED}_{p,o}\mathcal{ZA}_{a,j,k,o,t} - \left(1 - \mathcal{UA}_{a,j,k,m,t}\right)BIG \quad \forall (a,p) \in BOMA, j \in L_2, k \in K, m \in M, t \in T \tag{3.89}$$

$$\mathcal{KP}_{p,k,m,t} \geq \mathcal{QP}_{p,k,m}\mathcal{UP}_{p,k,m,t} \quad \forall\, p \in P, k \in K, m \in M, t \in T \tag{3.90}$$

$$\mathcal{JC}_{c,i,j,m,t} \geq \sum_{o\in OPT_C}\mathcal{LSC}_{c,i,j,m,t}\mathcal{EA}_{a,o}\mathcal{ZC}_{c,i,j,o,t} - \left(1 - \mathcal{UC}_{c,i,j,m,t}\right)BIG \quad \forall (c,a) \in BOMC, i \in L_1, j \in L_2, m \in M, t \in T \tag{3.91}$$

$$\mathcal{J}\mathcal{A}_{a,j,k,m,t} \geq \sum_{o \in OPT_A} \mathcal{LSA}_{a,j,k,m,t}\mathcal{ED}_{p,o}\mathcal{ZA}_{a,j,k,o,t} - \left(1 - \mathcal{UA}_{a,j,k,m,t}\right)BIG \qquad \forall (a,p) \in BOMA, j \in L_2, k \in K, m \in M, t \in T \tag{3.92}$$

$$\mathcal{JP}_{p,k,m,t} \geq \mathcal{LSP}_{p,k,m}\mathcal{UP}_{p,k,m,t} \qquad \forall\, p \in P, k \in K, m \in M, t \in T \tag{3.93}$$

$$\mathcal{GC}_{c,i,j,m,t} \geq \sum_{o \in OPT_C} \mathcal{QC}_{c,i,m}\mathcal{EA}_{a,o}\mathcal{ZC}_{c,i,j,o,t} - \left(1 - \mathcal{VC}_{c,i,j,m,t}\right)BIG \qquad \forall\, (c,a) \in BOMC, i \in L_1, j \in L_2, m \in M, t \in T \tag{3.94}$$

$$\mathcal{GA}_{a,j,k,m,t} \geq \sum_{o \in OPT_A} \mathcal{QA}_{a,j,m}\mathcal{ED}_{p,o}\mathcal{ZA}_{a,j,k,o,t} - \left(1 - \mathcal{VA}_{a,j,k,m,t}\right)BIG \qquad \forall (a,p) \in BOMA, j \in L_2, k \in K, m \in M, t \in T \tag{3.95}$$

$$\mathcal{WC}_{c,i,j,m,t} \geq \sum_{o \in OPT_C} \mathcal{LSC}_{c,i,j,m,t}\mathcal{EA}_{a,o}\mathcal{ZC}_{c,i,j,o,t} - \left(1 - \mathcal{VC}_{c,i,j,m,t}\right)BIG \qquad \forall (c,a) \in BOMC, i \in L_1, j \in L_2, m \in M, t \in T \tag{3.96}$$

$$\mathcal{WA}_{a,j,k,m,t} \geq \sum_{o \in OPT_A} \mathcal{LSA}_{a,j,k,m,t}\mathcal{ED}_{p,o}\mathcal{ZA}_{a,j,k,o,t} - \left(1 - \mathcal{VA}_{a,j,k,m,t}\right)BIG \qquad \forall (a,p) \in BOMA, j \in L_2, k \in K, m \in M, t \in T \tag{3.97}$$

$$\begin{aligned} &\mathcal{ZC}_{c,i,j,o,t} \in \{0,1\} \qquad \forall c \in C, i \in L_1, j \in L_2, o \in OPT_C, t \in T \\ &\mathcal{ZA}_{a,j,k,o,t} \in \{0,1\} \qquad \forall a \in SA, j \in L_2, k \in K, o \in OPT_A, t \in T \end{aligned} \tag{3.98}$$

$$\mathcal{KC}_{c,i,j,m,t}, \mathcal{KA}_{a,j,k,m,t}, \mathcal{KP}_{p,k,m,t}, \mathcal{JC}_{c,i,j,m,t}, \mathcal{JA}_{a,j,k,m,t}, \mathcal{JP}_{p,k,m,t}, \mathcal{GC}_{c,i,j,m,t}, \mathcal{G} \\ \mathcal{WC}_{c,i,j,m,t}, \mathcal{WA}_{a,j,k,m,t} \geq 0, \quad \forall c \in C, a \in SA, p \in P, i \in L_1, j \in L_2, k \in K, m \in M, t \in T \tag{3.99}$$

Constraints (3.38)-(3.42) are replaced by Constraints (3.69)-(3.73). In addition, Constraints (3.53), (3.54), (3.56), and (3.57) are replaced by Constraints (3.74) - (3.77), serving as a linkage between the two models. Subsequently, Constraints (3.78) and (3.79) are applied within the context of the first layer. Specifically, Constraints (3.78) are designed to determine the number of downstream nodes an upstream node supplies.

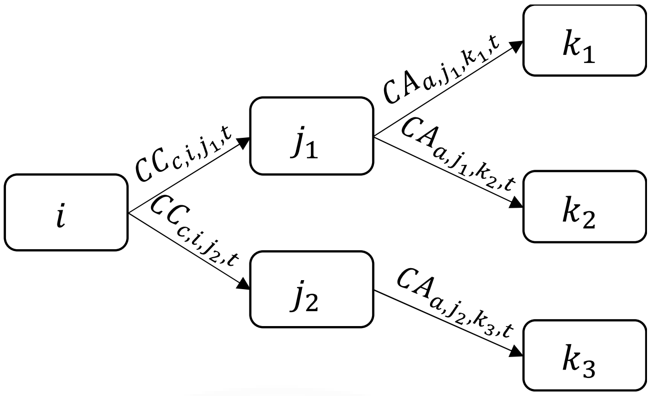



**Figure 3.4** Illustration of Constraint (3.78)

Consider Figure 3.4, the process of establishing a valid SS multiplier along an arc that links node $i \in L_1$ to node $j \in L_2$ involves a stepwise approach. The initial step determines the downstream sub-assembly nodes $j$ that $i$ supplies to, as well as identifying the specific final assembly manufacturers $k \in K$ that are served by node $j$. For instance, when considering the pair $i$ and $j_1$, the condition $\sum_{o\in O}\sum_{k\in D}\mathcal{A}_{a,j_1,k,o}\mathcal{ZC}_{c,i,j,o,t}$ must be greater than or equal to $\sum_{k\in K} CA_{a,j_1,k,t} - \left(1 - CC_{c,i,j_1,t}\right)BIG = 2$. This condition indicates that the SD multiplier for the arc $(i, j_1)$ must be chosen from the pool of multipliers that correspond to a total of two arcs. It is important to note that multiple multipliers could satisfy this condition. Therefore, Constraint (3.79) is introduced to deduce the correct SD multiplier for the arc $(i, j_1)$. A similar logic is extended to Constraints (3.80) and (3.81), which are to determine the appropriate multipliers for arcs within the second layer $L_2$.

Constraints (3.82) - (3.87) impose limitations such that if an arc is designated for holding SS, only a single multiplier can be selected for that particular arc. This constraint structure ensures consistency in the assignment of multipliers. Next, Constraints (3.88) - (3.97) are to compute the expected SS and shortage quantities for every arc connecting an upstream node to a downstream node. Finally, Constraints (3.98) and (3.99) pertain to binary variables and non-negativity, respectively.

# CHAPTER 4
# GENETIC ALGORITHM WITH RANK-BASED DECODING FOR THE INTEGRATED PROBLEM UNDER DIFFERENT SOURCING STRATEGIES

Integrating the IP with the NDP substantially increases complexity, mainly when dealing with multiple periods. Therefore, it is computationally impractical when attempting to solve large-scale instances through conventional commercial solvers. To tackle this challenge, this Chapter is to introduce a heuristic approach that handles the problem.

We opt for a GA as our solution approach, considering its suitability for addressing intricate problems, as established by Gen and Cheng (2000). The GA efficiently navigates the solution space of a problem, balancing exploration and exploitation to yield high-quality solutions within reasonable computational limits. Our proposed GA is developed for a three-layer network and is specifically designed to address two different product supply strategies: single-sourcing and multi-sourcing.

## 4.1 Chromosome

The selection of a chromosome design plays a critical role in determining the computational requirement of the GA. Our problem is composed of two distinct sub-problems: designing the structure of the network and establishing the locations and quantities of SS. In our method, the chromosome is responsible for encoding decisions related to both the opening of facilities and their respective capacity levels. Upon decoding, the chromosome facilitates the optimization of network flows and IP.

A chromosome consists of genes that are organized within an array. The array's size is determined by the number of nodes of the decoding layers, $|L_1| + |L_2|$. Within this array, two segments are present, known as sub-chromosomes $SC$, denoted as $SC^g, \forall g = \{1, 2\}$. A gene's position in a sub-chromosome signifies the location and capacity level, i.e., $V(i, w), \forall i \in L_1, w \in W$ or $V(j, w), \forall j \in L_2, w \in W$. Each gene, represented as $V(i, w)$ (or $V(j, w)$), is assigned a distinct rank value ranging from 1 to

$\mathcal{T}_{max}$, where $\mathcal{T}_{max} = |L_1| \times |W|$ (or $|L_2| \times |W|$) is the smallest rank to be selected. It should be noted that a chromosome represents the ND choices for the initial period.

The process of decoding involves managing binary variables within the NDP model, which includes variables such as $AC_{w,i,t}$ and $AA_{w,j,t}$. Specifically, the values attributed to these variables are determined by the rank values assigned to genes in a chromosome. By employing this approach, the NDP model is simplified into a conventional minimum cost flow problem.

To illustrate how the NDP can be encoded using a chromosome, consider a three-stage network consisting of eight potential component manufacturers, five potential subassembly manufacturers, and three predetermined final assembly manufacturers. In this context, the gene $V(1,0)$ represents the rank value associated with node 1 having the base capacity.

| | **(1,0)** | **(2,0)** | **(3,0)** | **…** | **(7,1)** | **(8,1)** | **(9,0)** | **(10,0)** | **…** | **(13,1)** |
|---|---|---|---|---|---|---|---|---|---|---|
| $t = 1$ | 16 | 9 | 8 | … | 14 | 10 | 2 | 9 | … | 1 |

**Figure 4.1** An example of a chromosome

## 4.2 Rank-based decoding

The decoding procedure begins from layer $L_2$, which is the closest to layer $K$, and progresses backward towards layer $L_1$. Specifically, the sub-chromosome $SC^2$ of the most downstream echelon is decoded first. The outcome of this first decoding process provides the values for binary variables $AA_{w,j,t}, \forall j \in L_2, w \in W,\ t \in T$. Following this, the same process is applied to decode the sub-chromosome $SC^1$ from layer $L_1$. This process yields the values for binary variables $AC_{w,i,t}, \forall i \in L_1, w \in W, t \in T$. After obtaining the values of all binary variables $AC_{w,i,t}$ and $AA_{w,j,t}$, these values are input into the MILP ND to define the network structure. For a visual representation of these concepts, refer to Figure 4.2, which presents an example of a three-layer network. The figure also shows the chromosome layout and provides information about the demand for a given period.

$C$:

| (SM1,0) | (SM2,0) | (SM3,0) | (SM1,1) | (SM2,1) | (SM3,1) | (M1,0) | (M2,0) | (M3,0) | (M1,1) | (M2,1) | (M3,1) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 3 | 5 | 6 | 1 | 2 | 6 | 3 | 1 | 5 | 4 |

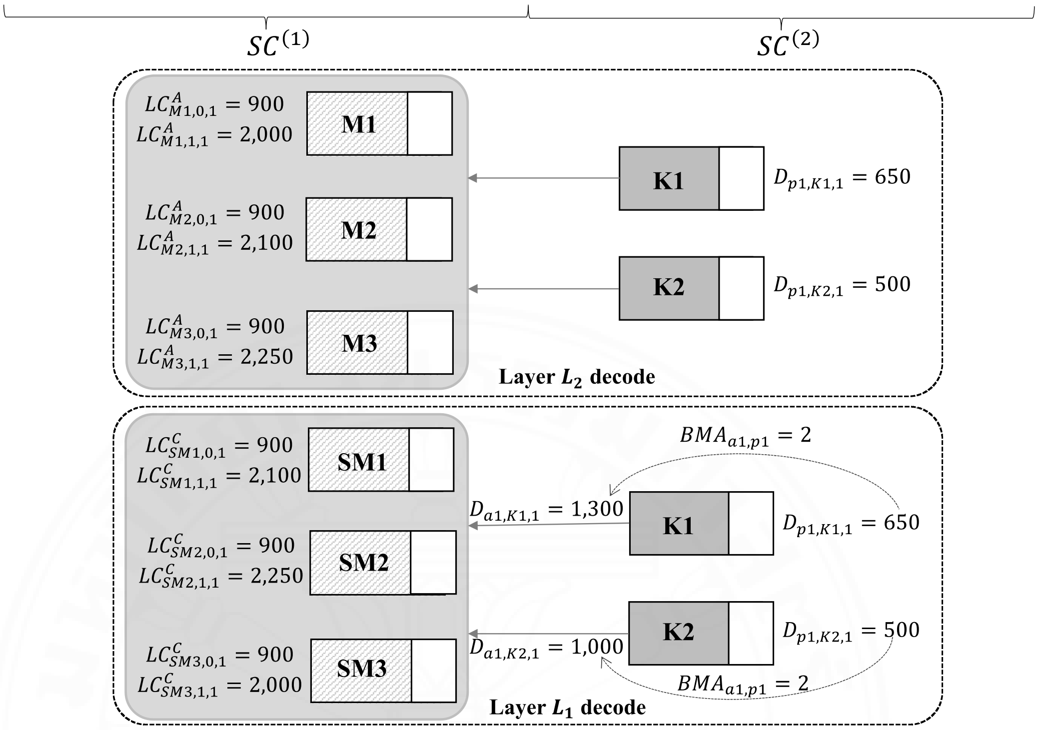



**Figure 4.2** Illustration of the decoding procedure

**Additional variable:**

$F_{a,j,w,k,t}$: Real variable, representing the proportion of subassembly $a \in SA$ supplied to final manufacturer $k \in K$ from subassembly manufacturer $j \in L_2$ holding capacity level $w \in W$ in period $t \in T$;

| |
|---|
| **Decoding procedure** |
| **Set** $g = 2$ |
| **While** $g > 0$: |
|     **Set** $\mathcal{T}_{max} = \lvert L_2 \rvert \times \lvert W \rvert$ |
|     **Set** $t = 1$ |
|     **Step 0:** |
|         **Initialization:** |
|             $G_{j,w}$: Priority value of gene $V(j,w), \forall\, j \in L_2, w \in W$. |
|             $LC_{j,w,t}$: Possible capacity levels $w \in W$ of node $j \in L_2$ in period $t \in T$. |

$D_{p,k,t}$: Demand for finished product $p \in P$ at the final assembly manufacturer $k \in K$ in period $t \in T$;

$TC_{j,t}$: Total capacity at node $j \in L_2$ in period $t \in T$;

$\Delta_{a,j,t}$: The inventory carry-over of subassembly $a \in SA$ at node $j \in L_2$ from the previous period that can be used to fulfill the current demand $t \in T$.

**While** $t \leq |T|$:

**Step 1:**

**If** $t = 1$: rank values of all the genes in sub-chromosome $SC^g$ are given.

**Set** $\Delta_{a,j,t} = 0, \forall a \in SA, j \in L_2$

**Set** $TC_{j,t} = 0, \forall j \in L_2$

**Set** $LC_{j,w,t} = \begin{cases} CAPA_{j,w}, \forall j \in L_2 & if\ w \in \{0\} \\ CAPA_{j,0} + CAPA_{j,w}, \forall j \in L_2 & if\ w \in W \backslash \{0\} \end{cases}$

Go to Step 2.

**Else If** $t > 1$:

**Set** $TC_{j,t} = TC_{j,t-1} + \sum_{w \in W} LC_{j,w,t-1} AA_{j,w,t-1}, \forall j \in L_2$

**If** $ZA_{j,t-1} = 1$:

**Set** $LC_{j,w,t} = CAPA_{j,w}, \forall j \in L_2, w \in W$

**Else:**

**Set** $LC_{j,w,t} = \begin{cases} CAPA_{j,w}, \forall j \in L_2 & if\ w \in \{0\} \\ CAPA_{j,0} + CAPA_{j,w}, \forall j \in L_2 & if\ w \in W \backslash \{0\} \end{cases}$

**Assign** rank values for the genes in sub-chromosome $SC^g$ based on the following conditions:

(1) Count the number of opened nodes from the previous period $ON = \sum_{j \in L_2} ZA_{j,t-1}$.

(2) Create two sets:

- Set of high rank values $HR = \{1, \ldots, (\mathcal{T}_{max} - ON \times |W|)\}$
- Set of low rank values $LR = \{(\mathcal{T}_{max} - ON \times |W|) + 1, \ldots, \mathcal{T}_{max}\}$

(3) **For** $j \in L_2$:

**If** $ZA_{j,t-1} = 1$:

- **Randomly assign** a rank value in set $HR$ to gene $V(j,w), \forall\, w \in W$.
- **Set** $G_{j,0} = \mathcal{T}_{max}$ for genes $V(j,0)$

**Else**:

- **Randomly assign** a rank value from set $LR$ to the remaining genes.

Go to Step 2.

**Step 2:** The proposed decoding procedure is formulated as an assignment problem, which focuses on allocating capacity levels to chosen facilities. The objective is to find an assignment to minimize the total rank. The formulation of our assignment problem is as follows.

$$Minimize \sum_{j\in L_2}\sum_{w\in W} G_{j,w}AA_{j,w,t} \tag{4.1}$$

**Multi-sourcing problem**

Subject to

There is only one capacity level to be selected for a node.

$$\sum_{w\in W} AA_{j,w,t} \leq 1, \forall j \in L_2 \tag{4.2}$$

The total capacity of all the nodes must be sufficient to fully satisfy the total demand of all customers, taking into account the tolerance of demand satisfaction.

$$\sum_{j\in L_2} TC_{j,t} + \sum_{j\in L_2}\sum_{w\in W} LC_{j,w,t}AA_{j,w,t} \geq \sum_{a\in SA}\sum_{p\in P}\sum_{k\in K} D_{p,k,t}BMA_{a,p} - \sum_{a\in SA}\sum_{j\in L_2}\Delta_{a,j,t} \tag{4.3}$$

The cumulative supply fraction of each item from all nodes to a customer must meet the demand of that customer.

$$\sum_{j\in L_2}\sum_{w\in W} F_{a,j,w,k,t} = 1, \qquad \forall a \in SA, k \in K \tag{4.4}$$

Any opened node is eligible as a supply node.

$$F_{a,j,w,k,t} \leq ZA_{j,t}, \qquad \forall a \in SA, j \in L_2, w \in W, k \in K \tag{4.5}$$

The capacity of a node cannot exceed the total amount that it supplies to downstream customers.

$$TC_{j,t} + \sum_{w \in W} LC_{j,w,t} AA_{j,w,t} \geq \sum_{a \in SA} \sum_{w \in W} \sum_{p \in P} \sum_{k \in K} D_{p,k,t} BMA_{a,p} F_{a,j,w,k,t} - \sum_{a \in SA} \Delta_{a,j,t}, \forall j \in L_2 \quad (4.6)$$

$$\sum_{w \in W} AA_{j,w,t} = ZA_{j,t} \qquad \forall j \in L_2 \qquad \text{If } t = 1 \quad (4.7)$$

$$ZA_{j,t} \geq ZA_{j,t-1} \qquad \forall j \in L_2 \qquad \text{If } t > 1 \quad (4.8)$$

Non-negativity

$$ZA_{j,t}, AA_{j,w,t} \in \{0,1\}, \forall j \in L_2, w \in W \quad (4.9)$$

$$F_{a,j,w,k,t} \in [0,1], \forall a \in SA, j \in L_2, w \in W, k \in K \quad (4.10)$$

**Single-sourcing problem**

$F_{a,j,w,k,t}$: binary variable, equals 1 if input item $a$ is supplied by subassembly manufacturer $j \in L_2$ to node $k \in K$, 0 otherwise;

Subject to

Constraints (4.2), (4.4) - (4.9)

The capacity of a node must not exceed demand for each item of a customer, taking into account the tolerance of the item satisfaction.

$$\sum_{j \in L_2} TC_{j,t} + \sum_{j \in L_2} \sum_{w \in W} LC_{j,w,t} F_{a,j,w,k,t} \geq \sum_{p \in P} D_{p,k,t} BMA_{a,p} - \sum_{a \in SA} \sum_{j \in L_2} \Delta_{t,a,j}, \forall a \in SA, k \in K \quad (4.11)$$

Tie up the two variables.

$$\sum_{a \in SA} \sum_{w \in W} \sum_{k \in K} F_{a,j,w,k,t} \geq ZA_{j,t}, \forall j \in L_2 \quad (4.12)$$

Non-negativity

$$F_{a,j,w,k,t} \in \{0,1\}, \forall a \in SA, j \in L_2, w \in W, k \in K \quad (4.13)$$

Go to Step 3.

**Step 3:**

Check and update $ZA_{j,t}$

$$ZA_{j,t} = min\,\{1, \sum_{w \in W} \sum_{\tau \in \{1,\ldots,t\}} AA_{j,w,\tau}\}, \forall\, j \in L_2 \tag{4.14}$$

**Set**:

$$\Delta_{a,j,t} = \min \left\{ \gamma, TC_{j,t} + \sum_{w \in W} LC_{j,w,t} AA_{j,w,t} - \sum_{p \in P} \sum_{w \in W} \sum_{k \in K} D_{p,k,t} BMA_{a,p} F_{a,j,w,k,t} \right\}, \forall a \in SA, j \in L_2 \tag{4.15}$$

Where $\gamma$ is a target remaining inventory at the end of the period which is limited by the storage capacity.

Go to Step 4.

**Step 4:**

**If** $t < |T|$, **set** $t = t + 1$, go to Step 1.

**Else**: **stop** the decoding for the current layer. Go to Step 5.

**Step 5:**

**If** $g > 1$, **set** $g = g - 1$, go to Step 0.

It should be noted that layer $L_1$ follows the same decoding procedure as described above. However, there are changes in the notation of parameters and decision variables. Specifically, the upstream node $j \in L_2$ changes to $i \in L_1$, the input item $a \in SA$ changes to $c \in C$, and the output item $p \in P$ changes to $a \in SA$.

For decoding $L_1$, the demand for finished product $a \in SA$ at the final manufacturer $k \in K$ in period $t \in T$, $D_{t,a,k}$, is computed as follows.

$$D_{a,k,t} = \sum_{p \in P} D_{p,k,t} BMA_{a,p}, \forall a \in SA | (a,p) \in BOMA, k \in K, t \in T \tag{4.16}$$

The parameters, including $BMA_{a,p}, CAPA_{j,w}$ are changed to $BMC_{c,a}, CAPC_{i,w}$. Concurrently, the variables $ZA_{j,t}, AA_{j,\mathrm{w},t}$ are changed to $ZC_{i,t}, AC_{i,\mathrm{w},t}$.

**Else**: stop the decoding process.

## 4.3 Example of rank-based decoding

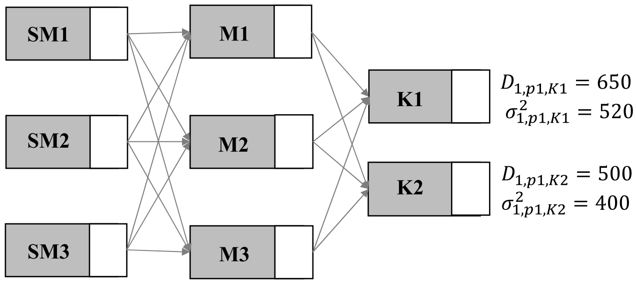



**Figure 4.3** Rank-based decoding example

Consider a network configuration comprising three layers: $L_1, L_2,$ and $K$. Nodes $K1$ and $K2 \in K$ demand a quantity of finished products $p1$. On the intermediate layer $L_2$, the subassembly manufacturing nodes are responsible for producing subassemblies $a1$. Meanwhile, nodes on layer $L_1$, referred to as component manufacturing nodes, produce and supply the required components $c1$ to downstream nodes. The production relationships are such that 2 units of subassembly $a1$ are needed to manufacture 1 unit of finished product $p1$, and 1 unit of component $c1$ is required to produce 1 unit of subassembly $a1$.

In the decoding process, the goal is to specify the number of operational facilities and their corresponding capacity levels. Facilities have the choice between two capacity levels: a base capacity (900 units upon initial opening) and an additional capacity for future expansion. Specific nodes within each layer possess the option to increase their capacities with different expansion limits. Specifically, the nodes $SM1, SM2,$ and $SM3$ can enhance their capacity by 1,200, 1,350, and 1,100 units, respectively. Similarly, the nodes $M1, M2,$ and $M3$ are possible to elevate their capacity levels by 1,100, 1,200, and 1,350 units, respectively.

Given the nature of single-sourcing and multi-sourcing strategies, the example is divided into two distinct scenarios. The decoding procedure initiates from the layer closest to the final manufacturing nodes, which are the nodes that receive customer demand. After completing the decoding for the current layer, the process proceeds backward to decode the upstream layer. This iterative decoding pattern continues until all layers within the network have undergone decoding.

Comparing the outcomes of single-sourcing and multi-sourcing decoding, Figure 4.4 Multi-sourcing and single-sourcing example provides a visual representation.

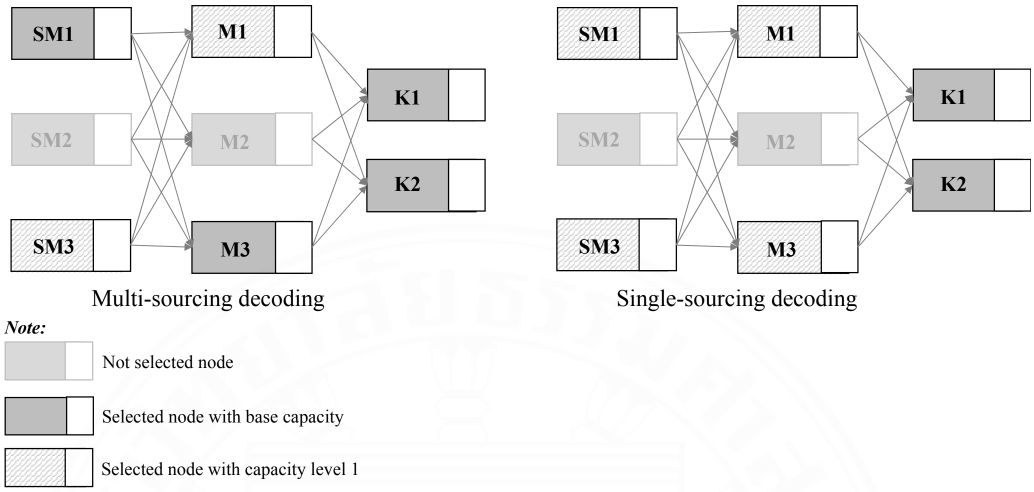



**Figure 4.4** Multi-sourcing and single-sourcing example

Additionally, Tables 4.1 and 4.2 exhibit the results derived from the decoding process for both multi-sourcing and single-sourcing strategies across three planning periods. Let us focus on the second stage of decoding. In that case, it is evident that the two sourcing strategies yield different capacity level suggestions for the opened facilities despite using the same chromosome. Specifically, when node $M3$ is opened, under the multi-sourcing strategy, we can opt for the base capacity to partially fulfill a fraction of the demand. In contrast, under the single-sourcing strategy and with a base capacity of 900 units, it is unfeasible to entirely meet the requirements of certain customers ($K1$ and $K2$) during the initial period, thus failing to satisfy the restrictions of the single-sourcing. Consequently, this necessitates the installation of additional capacity to meet these demands. The same procedure of decoding can be similarly extended to the first stage of decoding.

**Table 4.1** Layer $L_2$ decoding: Multi-sourcing

| $t$ | $V(j,w)$ | $j$ | Select? | $w$ | $\Delta_{a,j,t}$ | $TC_{j,t}$ | $LC_{j,w,t}$ | $ZA_{j,t}$ | $AA_{j,w,t}$ | Satisfied demand amount |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | [2, 6, 3, 1, 5, 4] | M1 | ✓ | 1 | 0 | 0 | 2,000 | 1 | 1 | 2,000 |
| | | M2 | - | - | 0 | 0 | - | - | - | - |
| | | M3 | ✓ | 0 | 0 | 0 | 900 | 1 | 1 | 300 |
| 2 | [7, 6, 7, 3, 5, 1] | M1 | - | - | 0 | 2,000 | 1,100 | 1 | - | 2,000 |
| | | M2 | - | - | 0 | 0 | - | - | - | - |
| | | M3 | - | - | 100 | 900 | 1,350 | 1 | - | 700 |
| 3 | [7, 5, 7, 1, 6, 3] | M1 | ✓ | 1 | 0 | 2,000 | 1,100 | 1 | 1 | 3,100 |
| | | M2 | - | - | 0 | 0 | - | - | - | - |
| | | M3 | - | - | 100 | 900 | 1,350 | 1 | - | 200 |

**Table 4.2** Layer $L_2$ decoding: Single-sourcing

| $t$ | $V(j,w)$ | $j$ | Select? | $w$ | $\Delta_{a,j,t}$ | $TC_{j,t}$ | $LC_{j,w,t}$ | $ZA_{j,t}$ | $AA_{j,w,t}$ | Satisfied demand amount |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | [2, 6, 3, 1, 5, 4] | M1 | ✓ | 1 | 0 | 0 | 2,000 | 1 | 1 | 1,300 |
| | | M2 | - | - | 0 | 0 | - | - | - | - |
| | | M3 | ✓ | 1 | 0 | 0 | 2,250 | 1 | 1 | 1,000 |
| 2 | [7, 6, 7, 3, 5, 1] | M1 | - | - | 100 | 2,000 | 1,100 | 1 | - | 1,200 |
| | | M2 | - | - | 0 | 0 | - | - | - | - |
| | | M3 | - | - | 100 | 2,250 | 1,350 | 1 | - | 1,500 |
| 3 | [7, 5, 7, 1, 6, 3] | M1 | - | - | 100 | 2,000 | 1,100 | 1 | - | 1,500 |
| | | M2 | - | - | 0 | 0 | - | - | - | - |
| | | M3 | - | - | 100 | 2,250 | 1,350 | 1 | - | 1,800 |

## 4.4 The proposed genetic algorithm

The GA process consists of six main steps for each generation as outlined below.

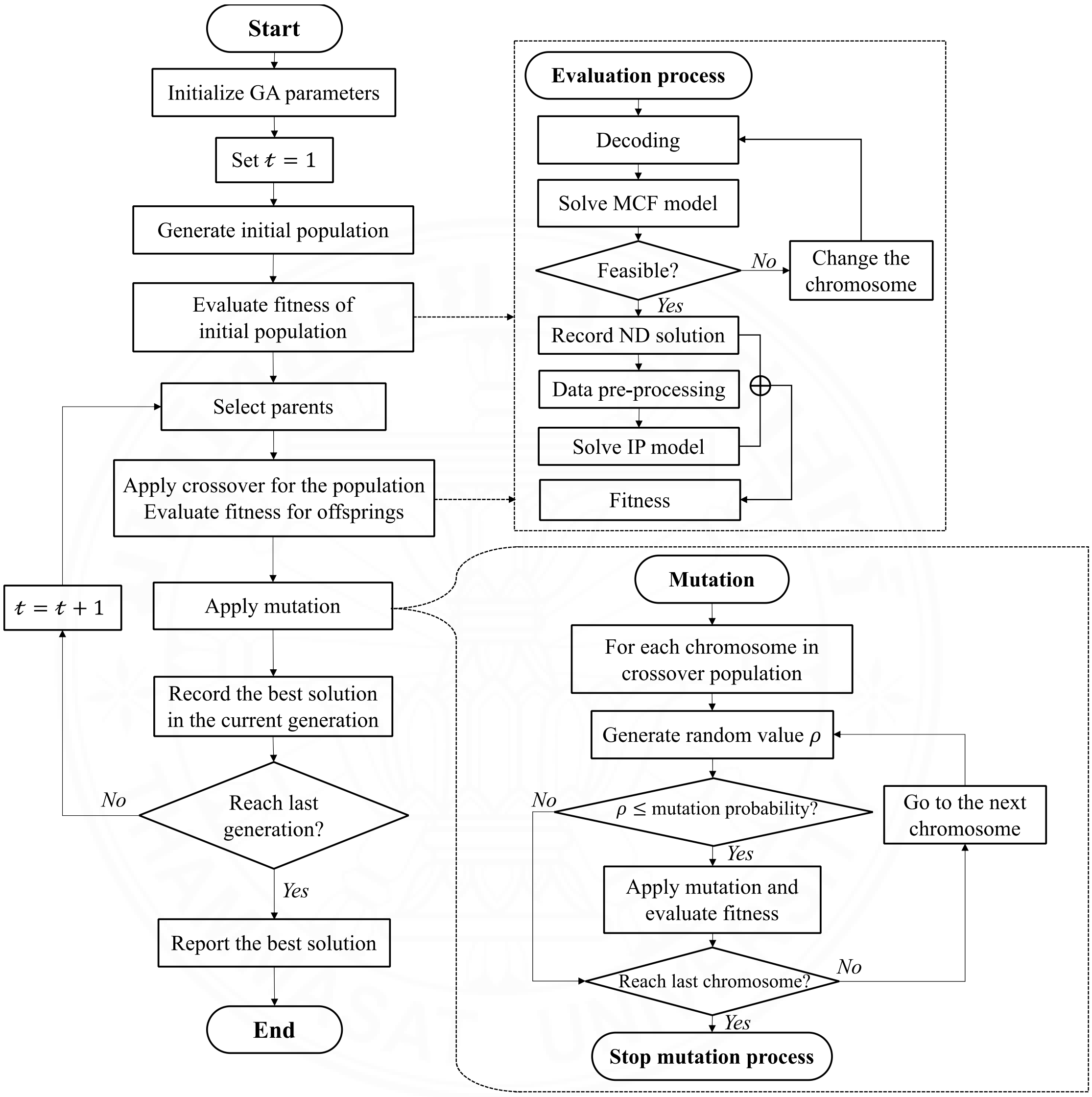



**Figure 4.5** The process of the proposed genetic algorithm

**Step 0: GA parameters setup**

- Define the mutation probability $m_p$.
- Set the selection rate $r\%$.
- Determine the population size $S$.

**Step 1: Initialization**

Set the current iteration $t = 1$.

Generate an initial generation comprising $\mathcal{C}|\mathcal{C} > S$ chromosomes, $\mathcal{C} = \{C^{(1)}, C^{(2)}, \dots, C^{(\mathcal{C})}\}$. The initial population is created randomly.

**Step 2: Fitness evaluation**

The fitness of a chromosome in the population is evaluated by identifying the value of the two objectives, including the ND and IP costs, denoted as $TC\left(C^{(c)}\right)$, where $c = \{1, 2, \dots, \mathcal{C}\}$.

Decode each chromosome using the rank-based decoding mechanism, as expressed in the section above.

Record the total ND cost of the feasible solution obtained after decoding. If the decoded result is infeasible, modify the chromosome until a feasible solution is achieved.

Once a feasible ND solution is achieved, preprocess essential parameters for the IP problem.

Calculate the expected SS amount and shortage for each arc during possible RTs based on the minimum-cost flow model (MCF).

Solve the IP model to determine optimal SS locations and amounts within the network. Concurrently, record the total SS cost.

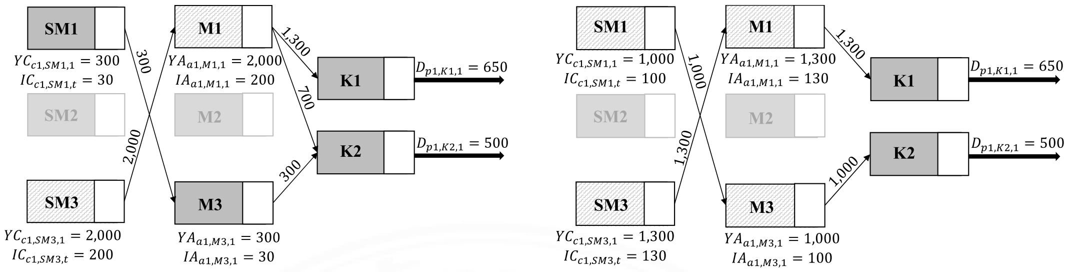
Combine the total ND and total SS costs to determine the fitness of the chromosome.

An example is presented to demonstrate the calculation of the fitness function. This example builds upon the decoding process outlined in the previous section. The subsequent expression outlines the steps from decoding to the final stage of computing the fitness.

**A. Minimum cost flow model (MCF):**

To achieve a balanced flow throughout the network and to minimize ND cost, three key variables are introduced: product flow, production quantity, and overall inventory amount. The assigned values to these variables play a critical role in maintaining equilibrium and optimizing the system cost. Figure 4.6 depicts the network structures along with the associated values of these three variables for two sourcing strategies, which are the results of solving the problems using the MCF model.

Additionally, the total costs of the NDP are calculated as 605,243.81 THB for the single-sourcing strategy and 523,077.23 THB for the multi-sourcing strategy. These total costs represent a cost component of the fitness function.




(a) Multi-sourcing decoding (b) Single-sourcing decoding

**Figure 4.6** Minimum-cost flow result

**B. Safety stock amount pre-processing**

As discussed in the previous section, the upper limit of demand for a particular product $p$ during a given time $m$ at node $i$ can be computed using the formula $m\mu_{p,i} - z_i\sigma_{p,i}\sqrt{m}$. In this equation, $\mu_i$ represents the mean cumulative demand at the node, $\sigma_{p,i}$ represents the SD of the demand distribution of product $p$ at that node, and $z_i$ denotes the safety factor corresponding to the specified CSL. In the formula, the term $z_i\sigma_{p,i}\sqrt{m}$ varies based on the value of $\sigma_{p,i}$. This variation is propagated across the SC network, originating from the final downstream nodes where the demand is directly observed. As this variability propagates throughout the network, different manufacturing nodes encounter differing degrees of variability. We introduce an approach to ascertain this variability across the network once the network structure is established.

- Supply ratio

Consider the results derived from the MCF model for both single-sourcing and multi-sourcing strategies. Each arc within the network indicates the quantity of products moving from an upstream node to a downstream node. Utilizing this quantity and the overall demand, a parameter referred to as the "supply ratio" can be computed. This supply ratio signifies the proportion of the downstream node's demand that is fulfilled by an upstream node through the corresponding arc. The computation of this supply ratio is as follows:

$$\varepsilon^{A}_{a,j,k,t} = \frac{XA_{a,j,k,t}}{\sum_{p\in P} BMA_{a,p} D_{p,k,t}}, \forall a \in SA, j \in L_2, k \in K, t \in T \tag{4.17}$$

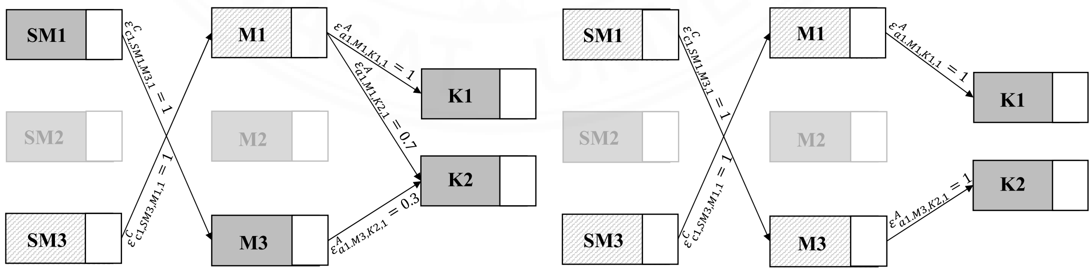
$$\varepsilon^{C}_{c,i,j,t} = \frac{XC_{c,i,j,t}}{\sum_{a\in SA} \sum_{k\in K} BMC_{c,a} YA_{a,j,t}}, \forall c \in C, i \in L_1, j \in L_2, t \in T \tag{4.18}$$

To illustrate, consider the flow denoted as $XA_{a1,M3,K2,1}$, which represents the movement of subassembly $a1$ from node $M3$ to node $K2$ during period 1. This flow exhibits different quantities in the context of single-sourcing and multi-sourcing strategies. Specifically, in the single-sourcing strategy, it delivers 1,000 units, whereas in the multi-sourcing context, it carries 300 units. Additionally, node $K2$ has a production requirement of $D_{p1,K2,1} = 500$ units for product $p1$, which relies on subassembly $a1$.

In the multi-sourcing strategy, the supply ratio associated with the arc $(M3, K2)$ is $\varepsilon^{A}_{1,a1,M3,K2} = \frac{XA_{a1,M3,K2,1}}{\sum_p BMA_{a,p} D_{p1,K2,1}} = \frac{300}{2\times 500} = 0.3$. This indicates that merely 30% of node $K2$'s demand is fulfilled by the flow originating from node $M3$.

Conversely, in the single-sourcing strategy, the supply ratio on the arc $(M3, K2)$ is $\varepsilon^{A}_{1,a1,M3,K2} = \frac{XA_{a1,M3,K2,1}}{\sum_p B^{A}_{a,p} D_{p1,K2,1}} = \frac{1{,}000}{2\times 500} = 1$. This implies that the complete demand of node $K2$ is met through the flow from node $M3$.

A comprehensive overview of the supply ratios for all arcs within the network is visually presented in Figure 4.7.




(a) Multi-sourcing decoding (b) Single-sourcing decoding

**Figure 4.7** Supply ratios of both strategies

Once the proportion of the demand is obtained, the next step is to calculate the SD of that specific proportion. This can be achieved through the following procedures:

- Multiply the SD by the supply ratio $\varepsilon^{C}_{c,i,j,t}$ (or $\varepsilon^{A}_{a,j,k,t}$) to determine the SD of the specific proportion. For subassembly $a$, the SD of the proportion $\varepsilon^{A}_{a,j,k,t}$, when the SD of product $p$ is known, is calculated as follows:

$$\sigma_{a,j,k,t} = BMA_{a,p}\varepsilon^{A}_{a,j,k,t}\sigma_{p,k}, \forall a \in SA, j \in L_2, k \in K, t \in T \tag{4.19}$$

- Compute the total SD of subassembly $a$ for a node, denoted as $\sigma_{a,j,t}$, by summing up the variances across all receiving nodes:

$$\sigma_{a,j,t} = \sqrt{\sum_{k \in K} \sigma^2_{a,j,k,t}}, \forall a \in SA, j \in L_2, t \in T \tag{4.20}$$

- Calculate the SD on each arc, $\sigma_{c,i,j,t}$, using the following formula:

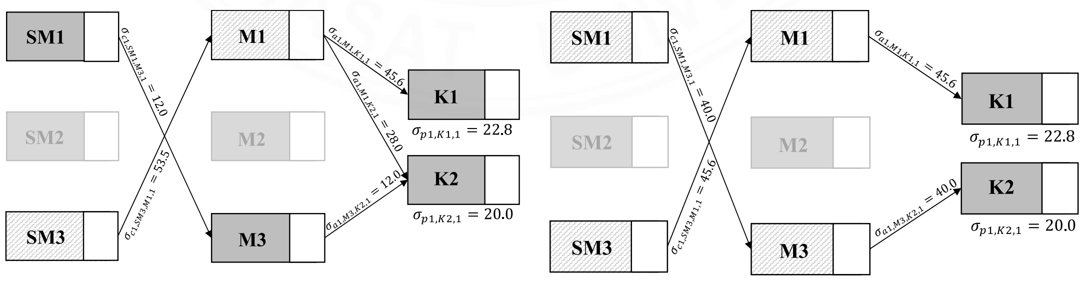
$$\sigma_{c,i,j,t} = BMC_{c,a}\varepsilon^{C}_{c,i,j,t}\sigma_{a,j,t}, \forall c \in C, i \in L_1, j \in L_2, t \in T \tag{4.21}$$

For instance, considering the previous example of $\varepsilon^{A}_{a1,M3,K2,1}$ for both single-sourcing and multi-sourcing strategies, the SD on the arc $(M3, K2)$ is computed as follows:

- For the multi-sourcing scenario: $2 \times 0.3 \times 20 = 12$.

- For the single-sourcing scenario: $2 \times 1 \times 20 = 40$.

These values are reflected in Figure 4.8.



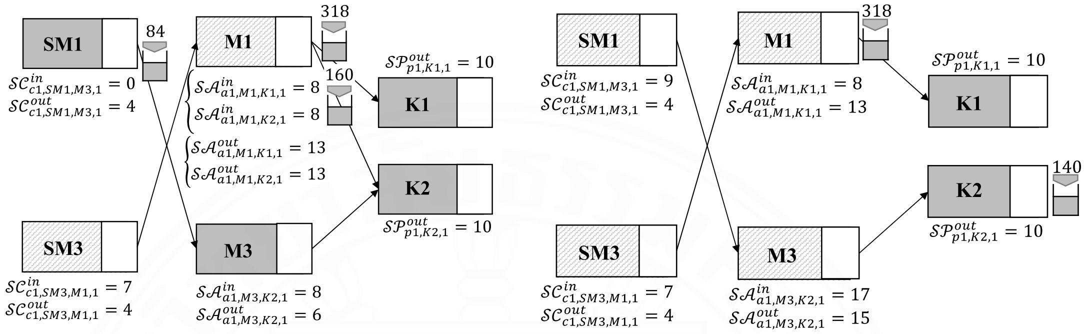

(a) Multi-sourcing decoding (b) Single-sourcing decoding

**Figure 4.8** SD result of both strategies

**C. IP Model**

**SS amount computation:** By using Equations (3.35) – (3.36), the expected SS amount and shortage can be obtained.

Once all the necessary information is collected, the IP model can be solved to determine the optimal SS positions. The arcs that accommodate SS reflect a make-to-stock strategy, while the remaining arcs follow a make-to-order approach. As illustrated in Figure 4.9, the single-sourcing strategy suggests two SS positions, whereas the multi-sourcing strategy recommends three positions.




(a) Multi-sourcing decoding (b) Single-sourcing decoding

**Figure 4.9** IP results for both sourcing strategies

After solving the IP model, the second cost component of the fitness function can be obtained. Specifically, the total safety stock costs are 150,475.60 THB and 184,097.16 THB for the multi-sourcing and single-sourcing scenarios, respectively.

**D. Fitness Function:**

Through the combination of the total ND cost computed in Part A and the IP cost obtained in Part C, the fitness values for both the multi-sourcing and single-sourcing strategies are achieved. Specifically, the fitness value for the multi-sourcing strategy stands at 673,552.83 THB, while for the single-sourcing strategy, the fitness is 789,340.97 THB. These fitness values represent the results achieved for the specific chromosome.

**Step 3: Selection**

In order to identify $S$ parents for generating offspring in the subsequent step, two mechanisms are utilized:

**Mechanism 1**: The top $r\%$ best-performing chromosomes from the entire population are initially chosen based on their fitness. These selected chromosomes then become participants in the crossover process.

**Mechanism 2**: The remaining $S - S \times r\%$ chromosomes are selected using the roulette wheel selection method. This method determines the chromosomes that will advance to the next step based on a probability value proportional to their fitness compared to the cumulative fitness of the entire population. This approach gives higher-fitness chromosomes a greater chance of being selected. The process for roulette wheel selection is outlined as follows:

- Determine the highest objective function value (highest fitness function): $TC_{max} = \max\{TC(C^{(c)})\}, \forall c = 1, 2, \dots, \mathcal{C}\}$
- Calculate the reversed fitness value for each chromosome: $F^{(c)} = TC_{max} - TC(C^{(c)}), \forall c = 1, 2, \dots, \mathcal{C}$
- Compute the normalized fitness value for every chromosome: $\delta^{(c)} = \frac{F^{(c)}}{\sum_{i=1}^{N} F^{(i)}}, \forall c = 1, 2, \dots, \mathcal{C}$
- Calculate the cumulative normalized fitness value by accumulating the normalized fitness values of each chromosome along with the fitness values of all previous chromosomes.
- Randomly generate a value $\varepsilon$ from a uniform distribution $U(0, 1)$. Then, select the chromosome with the cumulative normalized fitness closest to $\varepsilon$. Repeat this process until $S$ parents are selected.

By employing roulette wheel selection, chromosomes with lower fitness have a smaller portion of the roulette wheel allocated to them. Consequently, chromosomes with higher probabilities on the roulette wheel are more likely to be chosen for selection.

Through the combination of these two mechanisms, we achieve a balanced approach to selecting parents for the crossover process. This ensures the retention of the fittest chromosomes while also accommodating other chromosomes based on their fitness probabilities, thus promoting genetic diversity within the selected parents.

**Step 4: Crossover**

Crossover, in conjunction with selection, constitutes one of the fundamental operators of GA. These operators play an important role in the GA's functioning by enabling the generation of new offspring from parent chromosomes. Crossover entails the recombination of genetic material from two parent chromosomes and facilitates the exchange of high-quality gene segments. Various crossover operators have been developed, but this study focuses on two fundamental ones: one-point crossover and order crossover. These crossover operations are applied to each sub-chromosome.

One-point crossover, as depicted in Figure 4.10, is a straightforward technique widely employed in crossover methods. It involves randomly selecting a crossover point for each pair of sub-chromosomes, cutting them at that point, and exchanging the segments after the cut to create two new individuals.

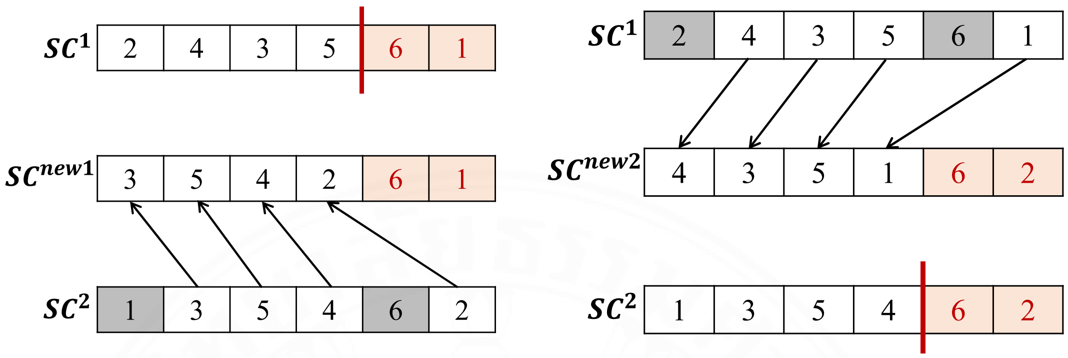



**Figure 4.10** One-point crossover

**One-point crossover procedure**

$$SC^{(1)} = \left[G_1^{(1)}, G_2^{(1)}, \dots, G_{\mathcal{T}_{max}}^{(1)}\right]$$

$$SC^{(2)} = [G_1^{(2)}, G_2^{(2)}, \dots, G_{\mathcal{T}_{max}}^{(2)}]$$

Generate a random number $x$ representing the cut point of a segment, where $x \leftarrow U(1, \mathcal{T}_{max})$.

All the genes starting from the cut point will be inherited by the new offsprings.

$SC^{(new_1)} \rightarrow G_x{}^{(new_1)}, \dots, G_{\mathcal{T}_{max}}{}^{(new_1)} = G_x{}^{(1)}, \dots, G_{\mathcal{T}_{max}}{}^{(1)}$

$SC^{(new_2)} \rightarrow G_x{}^{(new_2)}, \dots, G_{\mathcal{T}_{max}}{}^{(new_2)} = G_x{}^{(2)}, \dots, G_{\mathcal{T}_{max}}{}^{(2)}$

The priority values after selected will be removed to avoid the same priority appeared in the sub-chromosome. In addition, all the remaining gene positions of $SC^{(new1)}$ and $SC^{(new2)}$ are filled by the other parent as follows.

For $\theta = 1, \dots, x-1$:

For $\eta = 1, \dots, \mathcal{T}_{max}$:

$$G_\theta^{(new_1)} = G_\eta^{(2)} \text{ If } G_\eta^{(2)} \neq \{G_x{}^{(new_1)}, \dots, G_{\mathcal{T}_{max}}{}^{(new_1)}\}$$

$$G_\theta^{(new_2)} = G_\eta^{(1)} \text{If } G_\eta^{(1)} \neq \{G_x{}^{(new_2)}, \dots, G_{\mathcal{T}_{max}}{}^{(new_2)}\}$$

The order crossover, introduced by Davis (1991), serves as a fundamental permutation operator. In this operation, a consecutive segment of genes from the first parent is transferred to the offspring, and the remaining values of the child are placed in the same sequence as they appear in the second parent. The process of order crossover is visually represented in Figure 4.11.

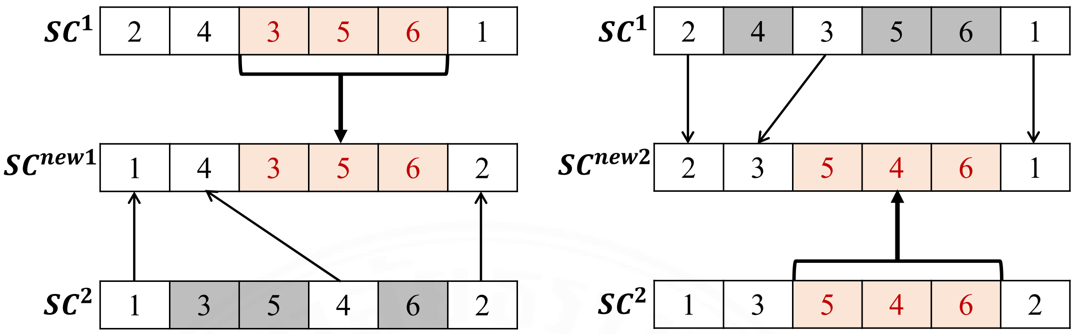



**Figure 4.11** Order crossover

| **Order crossover procedure** |
|---|
| Given |
| $$SC^{(1)} = \left[G_1^{(1)}, G_2^{(1)}, \dots, G_{\mathcal{T}_{max}}^{(1)}\right]$$ $$SC^{(2)} = [G_1^{(2)}, G_2^{(2)}, \dots, G_{\mathcal{T}_{max}}^{(2)}]$$ |
| Generate random numbers $x_1$ and $x_2$ representing the start point and end point of a swath, where $x_1 \leftarrow U(1, \mathcal{T}_{max}), x_2 \leftarrow U(x_1 + 1, \mathcal{T}_{max})$. |
| All the genes in the selected swath will be inherited by the new children. |
| $SC^{(new_1)} \rightarrow G_{x_1}{}^{(new_1)}, \dots, G_{x_2}{}^{(new_1)} = G_{x_1}{}^{(1)}, \dots, G_{x_2}{}^{(1)}$ |
| $SC^{(new_2)} \rightarrow G_{x_1}{}^{(new_2)}, \dots, G_{x_2}{}^{(new_2)} = G_{x_1}{}^{(2)}, \dots, G_{x_2}{}^{(2)}$ |
| The priority values after selected will be removed to avoid the same priority appeared in the sub-chromosome. In addition, all the remaining gene positions of $SC^{(new1)}$ and $SC^{(new2)}$ are filled by the other parent as follows. |
| For $\gamma = 1, \dots, x_1 - 1, x_2 + 1, \dots, \mathcal{T}_{max}$: |
|     For $\eta = 1, \dots, \mathcal{T}_{max}$: |
|         $G_\gamma^{(new_1)} = G_\eta^{(2)}$ If $G_\eta^{(2)} \neq \{G_{x_1}{}^{(new_1)}, \dots, G_{x_2}{}^{(new_1)}\}$ |
|         $G_\gamma^{(new_2)} = G_\eta^{(1)}$ If $G_\eta^{(1)} \neq \{G_{x_1}{}^{(new_2)}, \dots, G_{x_2}{}^{(new_2)}\}$ |
| Otherwise, $\eta \leftarrow \eta + 1$. Go to the next iteration. |

After the crossover phase, a population of size $2S$ individuals is generated from the initial population size of $S$.

**Step 5: Mutation**

GAs have to strike a balance between improving current solutions (through crossover) and exploring the solution space (via mutation) to increase the chances of discovering the optimal solution (Wong *et al.*, 2003). As a result, mutation serves as an essential operator to maintain population diversity.

Mutation acts as an operator that introduces random alterations to different chromosomes, promoting diversity within the population. Unlike crossover, which involves mixing genes between chromosomes, mutation generally involves modifying genes within a single chromosome.

In the proposed GA, we utilize inversion and swap mutations to make changes to a chromosome.

For each sub-chromosome $SC^{(\vartheta)} = \left[G_1^{(\vartheta)}, G_2^{(\vartheta)}, \dots, G_{\mathcal{T}_{max}}^{(\vartheta)}\right]$, a random probability $\rho$ is generated from a uniform distribution $U(0,1)$. If $\rho \leq m_p$, a mutation is performed using the following process. Otherwise, no mutation takes place.

Inversion mutation refers to a genetic anomaly in which a segment of the chromosome becomes detached and then attaches itself in the opposite orientation. An example of inversion mutation is provided below.

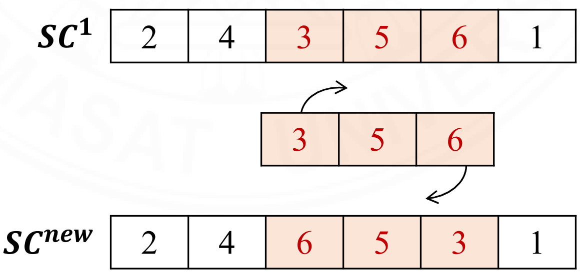



**Figure 4.12** Inversion mutation

| **Inversion mutation procedure** |
|---|
| Pick a random segment between $x_1 \leftarrow U(1, \mathcal{T}_{max})$ and $x_2 \leftarrow U(x_1 + 1, \mathcal{T}_{max})$. |
| Reverse the position of all the genes in the segment, where $G_\gamma^{(new)} = G_{x_1+x_2-\gamma}{}^{(\vartheta)}, \forall \gamma = \{x_1, \dots, x_2\}$. |

Swap mutation is another type of mutation employed in the proposed GA. This mutation involves selecting two genes of a chromosome and swapping their positions. An example of the swap mutation, where two genes located at randomly chosen positions are exchanged with each other, is presented as follows.

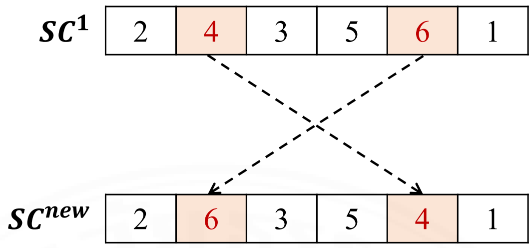



**Figure 4.13** Swap mutation

| **Swap mutation procedure** |
|---|
| Select a random position of a gene to swap $x \leftarrow U(1, \mathcal{T}_{max})$. |
| In addition, choose a priority value, i.e., $val \leftarrow U(1, \mathcal{T}_{max})$, to replace the value of gene $G_x^{(new)}$. |
| Finally, set $G_y^{(\vartheta)} = G_x^{(new)} \| G_y^{(\vartheta)} = val$ and $G_x^{(new)} = val$. |

After the mutation phase, all the chromosomes of the population are evaluated to identify the current best solution $TC^{(t)}$, and these values of $TC^{(t)}$ are then stored in an array.

Step 6: Following step 5, we proceed to evaluate the termination condition for the GA. Our stopping criterion is based on a prespecified maximum iteration $\mathcal{T}$. If the current generation $t = \mathcal{T}$, then report the best solution found $TC^{(best)} = \min\{TC^{(t)} | t = \{1, \ldots, \mathcal{T}\}$. Otherwise, we increment $t = t + 1$ and return to step 3 for the next generation.

| **Genetic algorithm procedure** |
|---|
| **Initialize** a population of a number of $\mathcal{R}$ chromosomes; Evaluate all the chromosomes; |
| **Repeat** |
| Select $S$ offsprings from $\mathcal{R}$ parents based on selection rate $r\%$ and roulette wheel; |
| Crossover to create $2S$ offsprings; |

Evaluate the $2S$ offsprings;

Mutate based on $m_p$;

Evaluate the mutated offsprings;

**Until** Stopping criteria

**Output** Best chromosome found.

# CHAPTER 5
# NUMERICAL EXAMPLE

## 5.1 Data description

This case study delves into a production network that specializes in manufacturing office furniture, specifically desk chairs and side chairs. The furniture pieces consist of three components: underframe, seat, and back. The network includes a total of sixteen nodes that span three layers. The first two layers comprise nodes engaged in the manufacturing process, while the third layer accommodates market nodes responsible for product distribution.

Specifically, the first layer $L_1$ encompasses two nodes dedicated to underframe manufacturing, three nodes for seat production, and another three nodes for back manufacturing. After completing the production phase in this layer, the components are forwarded to subassembly manufacturers located in layer $L_2$. These subassembly manufacturers are responsible for producing the subassemblies. Finally, the finished products are then assembled and dispatched to various market regions by layer $K$.

A key consideration is that to ensure consistent component quality, nodes engaged in finished goods manufacturing follow a single-sourcing strategy, wherein a single component manufacturer exclusively supplies each component. However, in scenarios where a multi-sourcing strategy is employed, this assumption can be relaxed, potentially allowing for more diverse sourcing options.

A visual representation of the network, along with its structure, is illustrated in Figure 5.1.

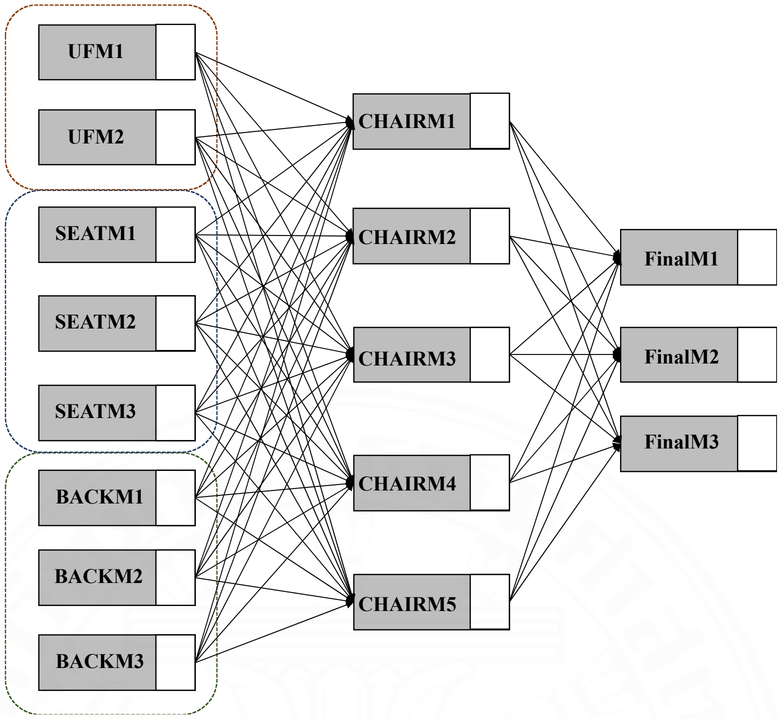



**Figure 5.1** Network structure for the chair manufacturing process

## 5.2 Result of the first approach: The integrated model

In this numerical example, we are considering a three-year planning horizon for our integrated MILP model. The relevant cost parameters, demand, and supply capacities are provided. Reflecting the company's market share in Thailand, we have estimated the demand for two specific types of chairs. Our objective is to demonstrate the effectiveness of our proposed model. Therefore, a comparison between two distinct approaches is conducted.

The first approach, known as the sequential approach, involves solving the ND and IP problems separately. The second approach, called the integrated approach, utilizes the integrated model outlined in Section 3 to address both ND and IP problems simultaneously. To derive results for both approaches, we employ the CPLEX 21.1 Solver. The three-period planning horizon problem instance in our case study contains a total of 60,822 decision variables. Among these, 39,810 are real variables, and 21,012 are binary variables. It takes approximately 222,934.55 seconds to obtain the optimal solution from CPLEX.

The customer demand is assumed to be normally distributed with specified average daily demand, $\mu_{p,k}$, and SD, $\sigma_{p,k}$. Our models are solved across various CSLs ranging from 99%, 99.1%, 99.2%,…, to 99.9%. The expected SS and shortage quantities are pre-computed based on Equations (3.64) - (3.67) for each individual product and corresponding CSL.

Additionally, the critical path, which signifies the longest route within the network, is determined. This critical path, established using the model outlined in Appendix B, provides the maximum feasible service time that a node in the network can commit to.

### 5.2.1 Comparison results of the MILP model

Figure 5.2 visually represents the total costs obtained from the two approaches across the specified range of CSLs.

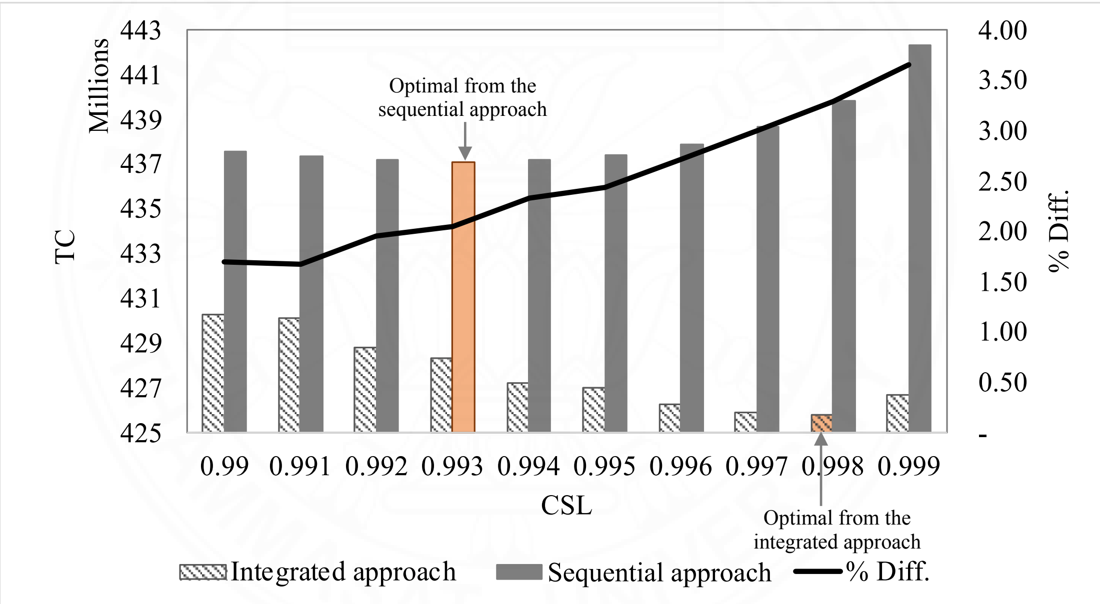



**Figure 5.2** Total costs of different CSLs between the two approaches

It is evident that the optimal CSLs recommended by these two approaches diverge. The proposed integrated approach designates the optimal CSL as 99.8%, which yields the lowest total cost across the considered range of CSLs. Conversely, the sequential approach identifies the optimal solution at a CSL of 99.3%. Notably, as the CSL increases beyond the optimal point determined by the sequential approach, the total cost experiences an escalation. In contrast, in the integrated approach, the total cost continually diminishes from 99% to 99.8% CSL until it reaches the optimal point. Beyond this point, at a CSL of 99.9%, the cost embarks on an ascent once again. Thus, the cost trends followed by these two approaches result in a substantial difference in percentage, as illustrated in the figure.

**Table 5.1** The optimal solution between the sequential approach and the integrated approach

| Model | Optimal CSL | NDP | IP | Total Cost (TC) |
|---|---|---|---|---|
| The sequential approach | 0.993 | 389,535,083.44 | 47,581,853.20 | **437,116,936.64** |
| The integrated approach | 0.998 | 396,560,213.94 | 29,247,466.39 | **425,807,680.33** |

Table 5.1 shows the optimal solutions achieved through our proposed integrated approach, compared with those derived from the sequential approach. The optimal CSLs lead to total costs of 425,807,680.33 THB and 437,116,936.64 THB for the integrated and sequential approaches, respectively. Although the difference is not substantial, there exists a slight variation in the ND costs of the two approaches due to the differing proposed network structures. Specifically, the integrated approach incurs a slightly higher ND cost but leads to a lower SS cost. Conversely, the sequential approach exhibits a lower ND cost, yet its total SS cost surpasses that of the integrated approach. The ND costs of the two approaches can be found in Table 5.2 The total network design cost between the integrated approach and the sequential approach.

Specifically, at 99.8% CSL, the integrated approach experiences a total SS cost of 29,247,466.39 THB, while the sequential approach's cost rises to 47,581,853.20 THB at 99.3% CSL. A comprehensive breakdown of the SS cost components can be found in Table 5.3. Despite incurring an additional 7,025,130.50 THB for ND, the integrated approach results in savings of 18,334,386.81 THB in SS positioning costs compared to the sequential approach. Consequently, the integrated approach achieves an overall lower total network cost, showcasing the advantage of solving the integrated model in comparison to the sequential approach.

In summary, the sequential approach attempts to optimize network configuration and subsequently locate SS results in a sub-optimal solution. In contrast, the proposed integrated model effectively balances the costs between ND and IP problems, leading to an optimal solution that outperforms the sequential approach.

Among the five cost components, the main differences come from the production management and the cycle inventory holding costs. Specifically, the production cost from the sequential approach is 23,887,337 THB lower than that of the integrated approach. This saving is enough to offset the increase of 12,597,565 THB in the cycle inventory holding cost. In total, these two cost components mainly contribute to the lower total ND cost from the sequential approach. Particularly, the saving occurs in the third period in the network.

Among the five cost components considered, the most significant differences arise from the production and cycle inventory holding costs. Precisely, the production cost incurred by the sequential approach is remarkably lower by 23,887,337 THB compared to the integrated approach. This cost reduction effectively counterbalances the concurrent increase of 12,597,565 THB in cycle inventory holding costs. In summary, these two primary cost components primarily drive the lower total ND cost observed in the sequential approach. Notably, this cost-saving becomes particularly evident in the third period of the network.

Referring to Table 5.3, a comprehensive examination of the IP cost components between the two strategies reveals noteworthy distinctions. In the sequential approach, the holding cost attributed to the SS of output items exceeds that of the integrated approach, while concurrently, the SS cost for input items is lower. Nevertheless, the overall SS holding cost within the integrated approach is mitigated by a considerable 14,835,929 THB.

Moreover, it is significant to note that the sequential approach advocates for a lower optimal CSL of 99.3%, indicating a high probability of encountering shortages and more expenses associated with unmet demand. Additionally, the sequential approach incurs elevated expenses in both SS holding and shortages, leading to a notably augmented total IP cost in contrast to the integrated approach.

**Table 5.2** The total network design cost between the integrated approach and the sequential approach

| Cost / Period | The sequential approach | | | | The integrated approach | | | | % Diff. |
|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **TC** | **1** | **2** | **3** | **TC** | **of TC** |
| Transportation | 11,079,185 | 10,450,721 | 10,747,868 | 32,277,774 | 9,510,961 | 10,150,040 | 10,686,631 | 30,347,632 | -6.36 |
| Fixed Opening | 105,778,500 | - | - | 105,778,500 | 102,484,000 | - | - | 102,484,000 | -3.21 |
| Fixed Installation | 49,440,000 | - | - | 49,440,000 | 50,400,000 | - | - | 50,400,000 | 1.90 |
| Production management | 25,527,130 | 26,945,136 | 27,484,039 | 79,956,305 | 30,612,216 | 32,107,886 | 41,123,539 | 103,843,642 | 23.00 |
| Inventory holding | 39,941,163 | 47,115,743 | 35,025,599 | 122,082,505 | 35,145,285 | 38,922,883 | 35,416,773 | 109,484,940 | -11.51 |
| Total network | 231,765,978 | 84,511,600 | 73,257,506 | 389,535,083 | 228,152,461 | 81,180,809 | 87,226,943 | 396,560,214 | 1.77 |

**Table 5.3** The total SS cost of between the integrated approach and the sequential approach

| Model | SS holding cost | | SS shortage cost | | Total IP cost |
|---|---|---|---|---|---|
| | **Output item** | **Input item** | **Output item** | **Input item** | |
| The sequential approach | 23,812,464 | 18,152,375 | 2,954,579 | 2,662,436 | 47,581,853 |
| The integrated approach | 3,520,135 | 23,608,775 | 330,183 | 1,788,374 | 29,247,466 |
| % Diff. | -474.42 | -18.79 | -75.89 | 19.06 | -71.96 |

**Table 5.4** Amount of SS in the network proposed by sequential and integrated approaches

| Approach | $t = 1$ | | | | | $t = 2$ | | | | | $t = 3$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **From Node** | **Item** | **SS of output** | **SS of input** | **To Node** | **From Node** | **Item** | **SS of output** | **SS of input** | **To Node** | **From Node** | **Item** | **SS of output** | **SS of input** | **To Node** |
| **Sequential** | **1** | c1 | - | 1,213 | **9** | **1** | c1 | - | 1,425 | **9** | **1** | c1 | - | 1,425 | **9** |
| | | c2 | - | 297 | | **9** | a1 | - | 1,561 | **14** | **9** | a1 | - | 1,561 | **14** |
| | **9** | a1 | 1,561 | - | **14** | **13** | a2 | - | 323 | | **13** | a2 | - | 323 | |
| | | a2 | 299 | - | | **9** | a1 | 1,239 | - | **15** | **9** | a1 | - | 1,239 | **15** |
| | **13** | a1 | 1,347 | - | **15** | **13** | a2 | - | 295 | | **13** | a2 | - | 295 | |
| | **9** | a2 | - | 272 | | **9** | a1 | 1,284 | - | **16** | **9** | a1 | 1,284 | - | **16** |
| | **9** | a1 | 1,284 | - | **16** | **13** | a2 | 293 | - | | **13** | a2 | 293 | - | |
| | | a2 | - | 280 | | | | | | | | | | | |
| **Integrated** | **12** | a1 | - | 1,180 | **14** | **12** | a1 | - | 1,180 | **14** | **12** | a1 | - | 1,180 | **14** |
| | **11** | a2 | - | 175 | | **11** | a2 | - | 175 | | | a2 | - | 226 | |
| | **12** | a1 | - | 875 | **15** | **12** | a1 | - | 875 | **15** | **11** | a1 | 758 | - | **15** |
| | **11** | a2 | 166 | - | | **11** | a2 | - | 166 | | **12** | a2 | - | 192 | |
| | **12** | a1 | - | 823 | **16** | **12** | a1 | - | 823 | **16** | **12** | a1 | - | 824 | **16** |
| | **11** | a2 | 207 | - | | **11** | a2 | 207 | - | | | a2 | 179 | - | |

Table 5.4 presents the SS quantities and their corresponding positions within the network. Each item in the network can potentially have two SS positions on an arc, including SS for output items retained post-processing by an upstream node and SS for input items prior to their processing by a downstream node. To demonstrate, consider the sequential approach's network when $t = 1$, along the arc connecting nodes 1 and 9, there are SS locations allocated for input items $c1$ and $c2$, constituting quantities of 1,213 and 297 units, respectively. Similarly, this logic extends to other nodes throughout the network.

**Table 5.5** Summary of total SS amounts of the two approaches

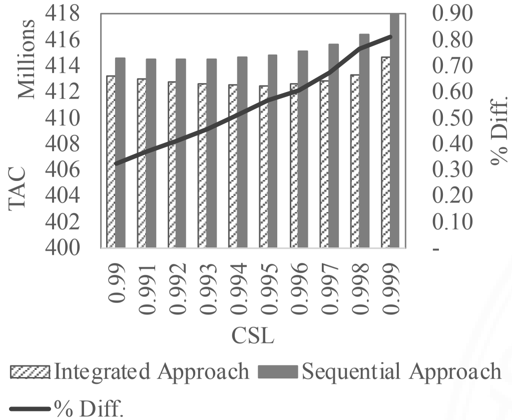
| Item | Sequential approach | Integrated approach |
|---|---|---|
| c1 | 4,062 | - |
| c2 | 297 | - |
| a1 | 12,359 | 8,518 |
| a2 | 2,674 | 1,695 |

The results also demonstrate that, during each period, the integrated approach's SS positions have shorter RTs when compared with the sequential approach. Additionally, the sequential approach proposes eight SS positions for the first period and seven for the subsequent periods, inclusive of additional positions interlinking nodes within the first and second layers. These shorter RTs and more rationalized SS sites result in diminished SS quantities. This reduction, in turn, leads to a decreased SS cost from the integrated approach. A summary of the cumulative SS quantities yielded by both approaches is presented in Table 5.5.

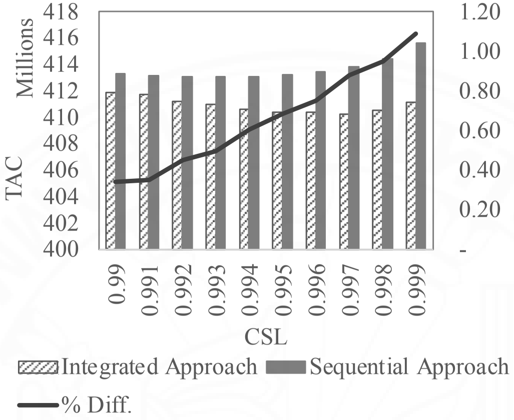
### 5.2.2 Sensitivity analysis

We conduct a sensitivity analysis to measure the impact of service time $R$ and the coefficient of variation $CV$ of demand on the optimal solutions. This experiment is performed by varying $R$ and $CV$ within defined ranges: $R$ spans from 6 to 10 periods, with an interval of 2, while $CV$ ranges from 0.25 to 0.75, with a step size of 0.25. Consequently, this exploration yields nine different $(R, CV)$ pairings, denoted as $\{(6, 0.25); (8, 0.25); \dots; (10, 0.75)\}$.

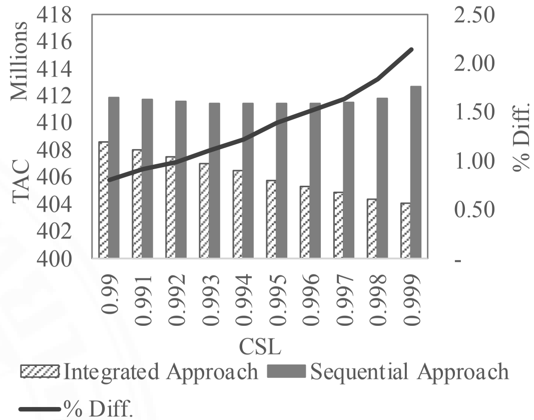
The outcome of this analytical exploration is visually illustrated in Figures 5.3 and 5.4.



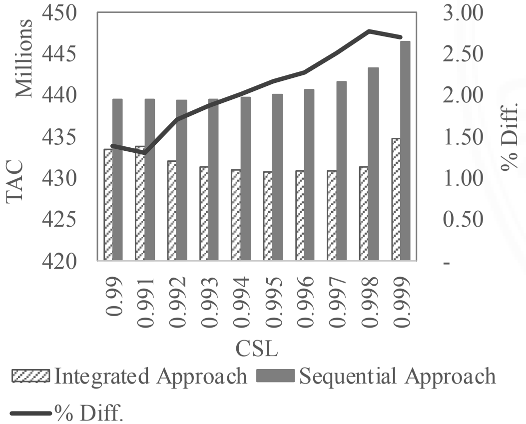

$R = 6, CV = 0.25$



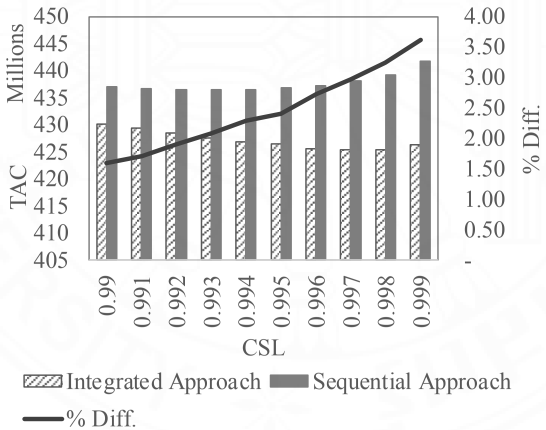

$R = 8, CV = 0.25$



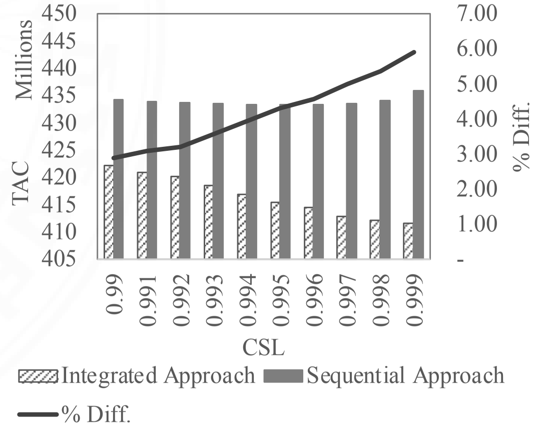

$R = 10, CV = 0.25$



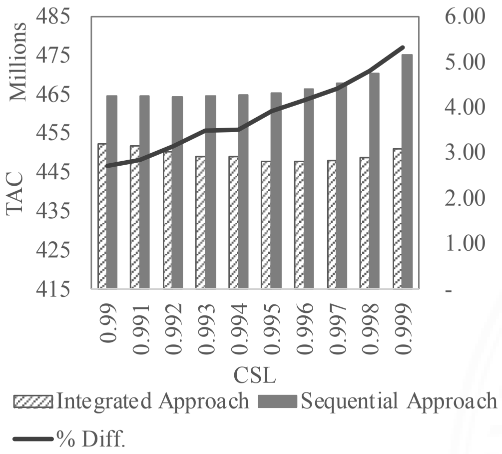

$R = 6, CV = 0.5$



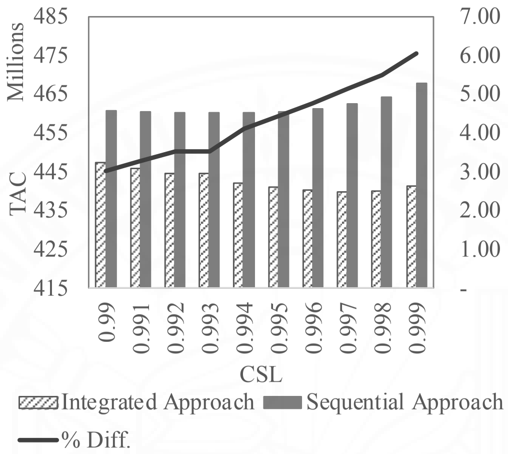

$R = 8, CV = 0.5$



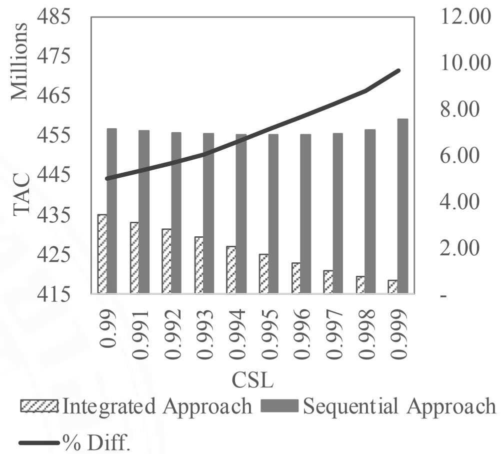

$R = 10, CV = 0.5$




$R = 6, CV = 0.75$




$R = 8, CV = 0.75$




$R = 10, CV = 0.75$

**Figure 5.3** Sensitivity analysis results of different combinations of $R - CV$

**Table 5.6** Optimal solutions at CSL of 99%

| Cost | ***R-CV*** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **6 - 0.25** | **8 - 0.25** | **10 - 0.25** | **6 - 0.5** | **8 - 0.5** | **10 - 0.5** | **6 - 0.75** | **8 - 0.75** | **10 - 0.75** |
| Transportation | 32,767,214 | 32,767,214 | 33,187,601 | 31,972,839 | 33,187,601 | 30,347,632 | 32,567,026 | 30,347,632 | 30,347,632 |
| Fixed Opening | 105,887,000 | 105,887,000 | 105,887,000 | 101,527,000 | 105,887,000 | 102,484,000 | 102,484,000 | 102,484,000 | 102,484,000 |
| Fixed Installation | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 |
| Production management | 94,521,442 | 94,521,442 | 94,545,483 | 102,861,476 | 94,545,483 | 103,843,642 | 101,594,470 | 103,843,642 | 103,843,642 |
| Inventory holding | 109,930,118 | 109,930,118 | 110,075,196 | 110,133,375 | 110,075,196 | 109,484,940 | 110,349,611 | 109,484,940 | 109,484,940 |
| **Total network** | **393,505,773** | **393,505,773** | **394,095,279** | **396,894,690** | **394,095,279** | **396,560,214** | **397,395,108** | **396,560,214** | **396,560,214** |
| SS holding cost of output items | 9,472,193 | 4,609,334 | 2,078,011 | 15,386,828 | 18,684,543 | 3,793,222 | 21,080,266 | 16,665,703 | 806,945 |
| SS holding cost of input items | 5,833,105 | 8,432,253 | 6,301,340 | 11,283,155 | 6,380,314 | 8,612,650 | 19,227,238 | 15,807,148 | 17,801,862 |
| SS shortage cost of output items | 2,740,157 | 1,972,371 | 1,282,066 | 5,584,006 | 8,547,707 | 4,033,526 | 7,755,257 | 9,554,884 | 858,066 |
| SS shortage cost of input items | 1,660,340 | 3,352,531 | 4,871,700 | 4,405,627 | 2,401,854 | 9,158,270 | 6,852,012 | 8,718,105 | 18,929,628 |
| **Total IP cost** | **19,705,796** | **18,366,490** | **14,533,117** | **36,659,617** | **36,014,418** | **25,597,668** | **54,914,773** | **50,745,840** | **38,396,501** |
| **Total** | **413,211,569** | **411,872,262** | **408,628,396** | **433,554,307** | **430,109,697** | **422,157,882** | **452,309,881** | **447,306,054** | **434,956,715** |

**Table 5.7** Optimal solutions at CSL of 99.5%

| Cost | *R-CV* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **6 - 0.25** | **8 - 0.25** | **10 - 0.25** | **6 - 0.5** | **8 - 0.5** | **10 - 0.5** | **6 - 0.75** | **8 - 0.75** | **10 - 0.75** |
| Transportation | 32,767,214 | 32,767,214 | 33,187,600 | 30,347,632 | 31,972,839 | 30,347,632 | 30,347,632 | 30,347,632 | 30,347,632 |
| Fixed Opening | 105,887,000 | 105,887,000 | 105,887,000 | 102,484,000 | 101,527,000 | 102,484,000 | 102,484,000 | 102,484,000 | 102,484,000 |
| Fixed Installation | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 | 50,400,000 |
| Production management | 94,521,442 | 94,521,442 | 94,545,4823 | 103,843,642 | 102,861,476 | 103,843,642 | 103,843,642 | 103,843,642 | 103,843,642 |
| Inventory holding | 109,930,118 | 109,930,118 | 110,075,196 | 109,484,940 | 110,133,375 | 109,484,940 | 109,484,940 | 109,484,940 | 109,484,940 |
| **Total network** | **393,505,773** | **393,505,773** | **394,095,279** | **396,560,214** | **396,894,690** | **396,560,214** | **396,560,214** | **396,560,214** | **396,560,214** |
| SS holding cost of output items | 4,248,579 | 7,731,749 | 1,962,452 | 24,843,867 | 13,609,006 | 4,985,730 | 18,195,336 | 27,079,441 | 4,382,805 |
| SS holding cost of input items | 12,626,877 | 6,607,427 | 6,278,864 | 4,775,645 | 10,293,565 | 6,079,443 | 26,233,932 | 8,875,852 | 12,378,054 |
| SS shortage cost of output items | 391,616 | 1,324,447 | 1,268,031 | 3,725,129 | 3,516,405 | 3,635,589 | 2,950,810 | 6,443,044 | 3,926,530 |
| SS shortage cost of input items | 1,702,852 | 1,220,176 | 2,205,555 | 840,442 | 2,269,457 | 4,209,686 | 3,897,546 | 2,080,146 | 7,738,059 |
| **Total IP cost** | **18,969,923** | **16,883,799** | **11,714,903** | **34,185,083** | **29,688,433** | **18,910,448** | **51,277,625** | **44,478,483** | **28,425,449** |
| **Total** | **412,475,696** | **410,389,571** | **405,810,182** | **430,745,297** | **426,583,124** | **415,470,662** | **447,837,839** | **441,038,697** | **424,985,663** |

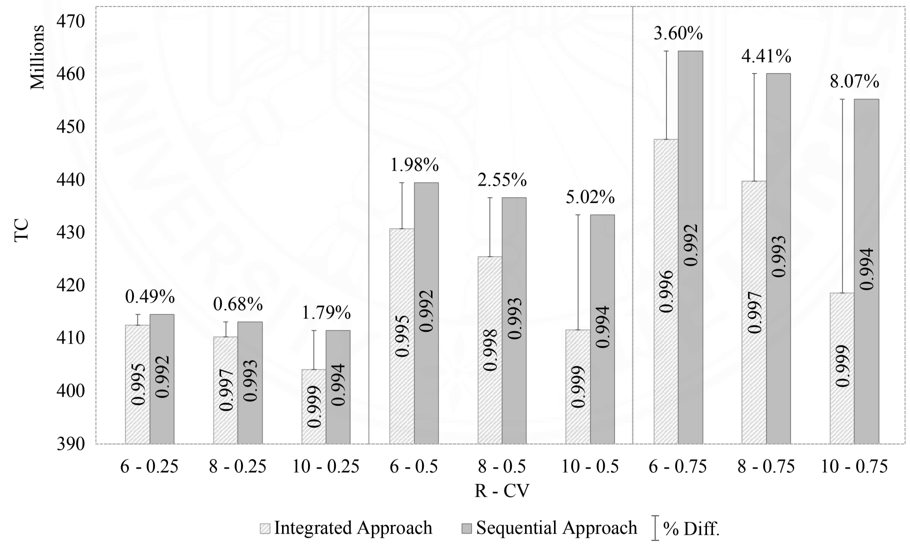



**Figure 5.4** Optimal solutions of different combinations of $R - CV$

Figure 5.3 graphically presents the impact of the nine combinations of $(R, CV)$ across a range of ten CSLs, spanning from 99% to 99.9%. From this representation, it indicates that an increase in $R$ corresponds to a reduction in the required SSs, consequently resulting in diminished total SS costs. The positive proportional relationship between the total annual SS cost and $CV$ becomes apparent, indicating that an increase in the SD results in escalated total SS costs due to higher demand uncertainty. This relationship is visually demonstrated in Figure 5.4.

Within this interaction, the CSL operates as a critical factor, representing a trade-off between SS holding and shortage costs. When holding the same $CV$, elevating $R$ (thus extending customer tolerance) gives a dual advantage: it reduces the overall SS cost and elevates the optimal CSL. An example of this phenomenon is noted when examining $CV = 0.25$. Here, as $R$ escalates from 6 to 8 within the sequential approach, the optimal CSL shifts from 99.2% to 99.4%, while the integrated approach proposes a shift from 99.5% to 99.9%.

Figure 5.4 underscores the broad gap in optimal CSL between the two approaches as $R$ remains constant and $CV$ increases. For instance, when $CV = 0.25$ and $R$ transitions from 6 to 8, the gap expands from 0.3% (99.5% - 99.2%) to 0.4% (99.7% - 99.3%) and further to 0.5% (99.9% - 99.4%). Similar trends are observed for $CV = 0.5$ and $CV = 0.75$. The percentage of improvement (% Diff.) signifies this expanding gap in optimal CSL with increasing $R$, resulting in more substantial savings in the total SS cost.

The sensitivity analysis result also reveals that there is an influence of safety stock decisions on the network structure. It is demonstrated that varying $R$-$CV$ values lead to distinct safety stock placement decisions, subsequently influencing choices regarding the network structure with the goal of minimizing the overall network cost. The changes in network structure can be seen in Table 5.6 and Table 5.7 by looking into the fixed opening and transportation cost components. Specifically, with the same CSL, i.e., 99% and 99.5%, the facility-opening decision and network flows may change across different values of $R$-$CV$. This finding reveals the relationship among committed service time, demand variability, safety stock, and network design decisions.

## 5.3 Result of the second approach: Genetic algorithm

In order to address a more extensive problem, considering a ten-period planning horizon, we employ the GA introduced in Chapter 4. The GA is developed using Python 3.7. Initially, a comparative analysis is conducted, wherein the GA results are compared with optimal solutions obtained from CPLEX. This comparative evaluation is performed on medium-scale single-sourcing problems spanning a three-period planning horizon. This comparison allows us to evaluate the effectiveness and efficiency of the GA in tackling more complex and extensive SC optimization challenges, providing insights into its performance compared to optimal solutions achieved through the CPLEX solver.

After this, our experimentation extends to both single-sourcing and multi-sourcing strategies, where the challenges are more substantial, involving a ten-period planning horizon. The GA is then used to address these larger problem instances. In the evaluation process, the performance of the GA is evaluated using upper bounds derived from CPLEX.

### 5.3.1 Comparison results of the GA

The GA in the initial experiment is configured to incorporate both selection mechanisms outlined in Chapter 4. In addition, the crossover operation is executed using the order crossover method, while swap mutation technique is employed in the mutation process. The GA's population comprises 20 chromosomes, which are generated randomly. The GA iterates through 200 generations and employs a mutation rate of $\rho = 0.1$. Additionally, for Mechanism 1 of the selection process in Step 3, the selection rate is set to $r = 0.5$.

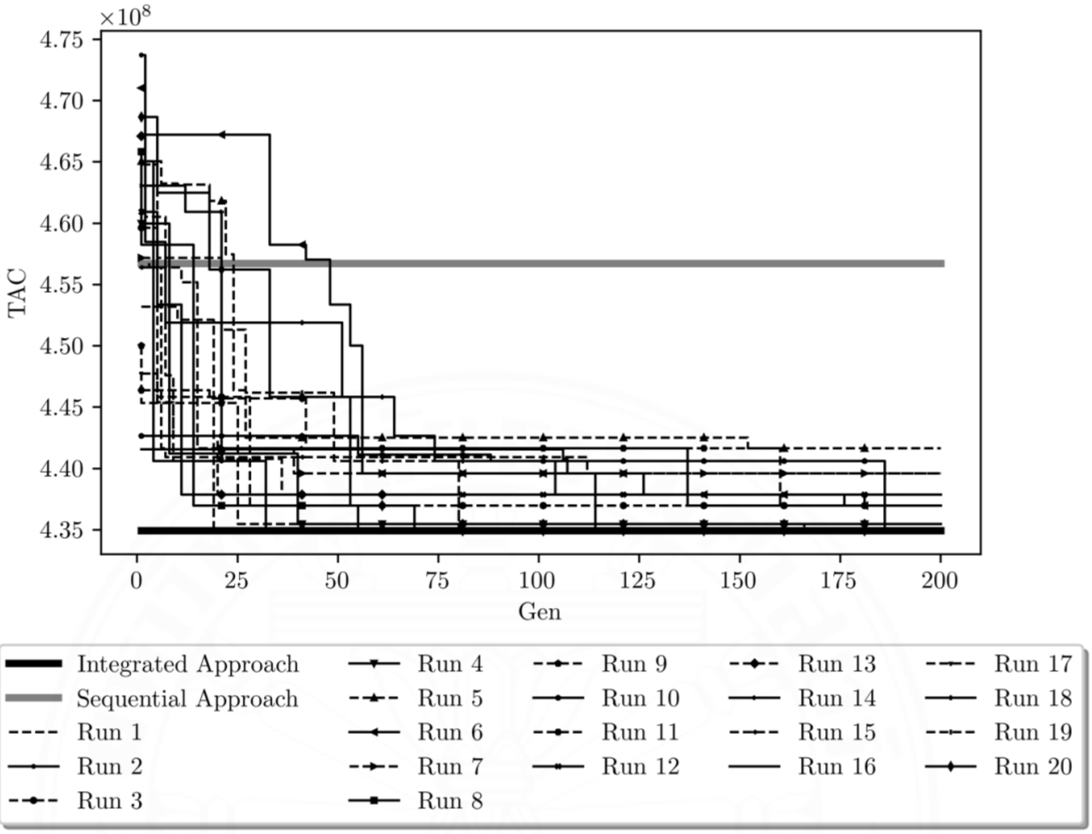



**Figure 5.5** GA results – single-sourcing strategy with three-period planning horizon

Optimal solutions from the sequential and integrated approaches are compared with the results from GA. Figure 5.5 visually represents the results of various runs of the GA in comparison with the two approaches in the context of a three-period planning horizon. It is evident that the solutions generated by the GA exhibit significant improvement within the initial 100 generations, after which the rate of improvement slows down during the subsequent generations. All runs of the GA outperform the solutions from the sequential approach. Moreover, among twenty GA runs, eight of them managed to achieve the optimal solution from the integrated approach (runs 1, 2, 8, 9, 10, 16, 19, and 20). On the contrary, the least favorable GA solution is observed in run 5. The solution obtained by the GA in run 5 after 200 generations is 441,654,481.71 THB, equivalent to a 1.54% gap compared to the optimal solution derived from the integrated approach. In addition, in terms of computational time, the proposed GA also outperforms the performance of the commercial solver. Specifically, the GA takes approximately 7,908.91 seconds to finish 200 generations, while CPLEX obtains the solution after 19,307.72 seconds.

Table 5.8 provides a comprehensive comparison of solutions among the three approaches. It is evident that the proposed GA effectively captures the trade-off between ND and IP problems, similar to the integrated approach. When it comes to ND cost, both the integrated approach and the proposed GA produce the same network structure, resulting in a higher network cost when compared to the sequential approach.

Comparing the components of IP costs across the three approaches, the sequential approach exhibits notably higher holding costs for SS but lower costs associated with SS shortages in comparison to the integrated approach and the GA. However, both the integrated approach and the GA managed to achieve an overall reduction in SS holding costs by a substantial amount of 28,786,790 THB. This substantial reduction in holding costs manages to offset the higher SS shortage costs incurred by the integrated approach and the GA.

In the context of the solutions provided by the integrated approach and the GA, even though the network structure remains consistent, there are minor differences in the total SS costs due to decisions about whether nodes keep input or output items. It should be noted that the IP model assumes equal unit holding and shortage costs for both input and output items at a given node. This assumption implies that keeping the same quantity at a node will yield a constant SS cost, regardless of whether input or output items are retained. Consequently, although there may be variations in the SS components between the integrated approach and the GA, the overall total SS cost remains unaffected.

**Table 5.8** Comparison of the solutions among the three approaches

| Cost \ Period | The sequential approach | | | | The integrated approach | | | | Genetic algorithm | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **Total** | **1** | **2** | **3** | **Total** | **1** | **2** | **3** | **Total** |
| Transportation | 11,079,185 | 10,450,721 | 10,747,868 | 32,277,774 | 9,510,961 | 10,150,040 | 10,686,631 | 30,347,632 | 9,510,961 | 10,150,040 | 10,686,631 | 30,347,632 |
| Fixed Opening | 105,778,500 | - | - | 105,778,500 | 102,484,000 | - | - | 102,484,000 | 102,484,000 | - | - | 102,484,000 |
| Fixed Installation | 49,440,000 | - | - | 49,440,000 | 50,400,000 | - | - | 50,400,000 | 50,400,000 | - | - | 50,400,000 |
| Production management | 25,527,130 | 26,945,136 | 27,484,039 | 79,956,305 | 30,612,216 | 32,107,886 | 41,123,539 | 103,843,642 | 30,612,216 | 32,107,886 | 41,123,539 | 103,843,642 |
| Cycle inventory holding | 39,941,163 | 47,115,743 | 35,025,599 | 122,082,505 | 35,145,285 | 38,922,883 | 35,416,773 | 109,484,940 | 35,145,285 | 38,922,883 | 35,416,773 | 109,484,940 |
| Total network | 231,765,978 | 84,511,600 | 73,257,506 | 389,535,083 | 228,152,461 | 81,180,809 | 87,226,943 | 396,560,214 | 228,152,461 | 81,180,809 | 87,226,943 | 396,560,214 |
| SS holding output item | 14,834,470 | 11,391,394 | 6,400,028 | 32,625,891 | 399,478 | 407,467 | - | 806,945 | 2,369,413 | - | 404,212 | 2,773,626 |
| SS holding input item | 2,784,925 | 6,352,214 | 12,053,324 | 21,190,462 | 5,594,622 | 5,706,514 | 6,500,726 | 17,801,862 | 3,624,686 | 6,113,982 | 6,096,514 | 15,835,182 |
| SS shortage output item | 3,332,563 | 2,507,747 | 1,584,524 | 7,424,835 | 424,785 | 433,281 | - | 858,066 | 2,519,518 | - | 429,819 | 2,949,338 |
| SS shortage input item | 978,047 | 1,931,629 | 3,032,427 | 5,942,103 | 5,949,047 | 6,068,028 | 6,912,554 | 18,929,628 | 3,854,314 | 6,501,309 | 6,482,734 | 16,838,356 |
| Total SS | 21,930,004 | 22,182,984 | 23,070,303 | 67,183,291 | 12,367,932 | 12,615,290 | 13,413,280 | 38,396,501 | 12,367,932 | 12,615,290 | 13,413,280 | 38,396,501 |
| Total | | | | 456,718,374 | | | | 434,956,715 | | | | 434,956,715 |

In the next section, the problem is extended to encompass a longer planning horizon that spans ten periods. Multiple runs of the GA using a set of parameter settings are employed to solve this expanded problem. Through an exploration of various parameter combinations, our objective is to identify the GA settings that offer the most promising chances for achieving superior solutions.

### 5.3.2 GA parameters tunning

The effectiveness of the proposed GA can be influenced by the configuration of its parameters, including mutation probability, mutation, crossover operators, population size, selection rate, and number of generations. Different combinations of parameter values can lead to varying performance regarding solution quality and convergence speed. The process of determining the most suitable parameter values in advance is referred to as the parameter tuning problem.

In our study, we address this parameter tuning problem through the utilization of a design of experiment, specifically employing a full factorial design. This type of experimental design enables us not only to assess the individual impacts of each parameter and operator but also to examine the potential interactions among them. To evaluate and optimize the performance of the GA, we carry out an experiment focused on the multi-sourcing problem. The full factorial design comprises four key factors, each tested with a range of settings to analyze their effects thoroughly. The four considered factors are listed as follows:

- Crossover operator: One-point crossover and order crossover
- Mutation probability: 0.1 and 0.3
- Mutation operator: inversion and swap mutation
- Selection rate: 0.1 and 0.5

Furthermore, we incorporate four center points as intermediate levels for both mutation probability and selection rate. These center points are utilized across all combinations of crossover and mutation operators. To enhance the robustness of our analysis, we perform each run five times, resulting in a total of 100 runs. Comprehensive details regarding the results of these five replicate runs can be found in Appendix C.

The statistical analysis of the experiment, including relevant tables and a graphical representation, is presented in Table 5.9, as well as in Figure 5.6. Notably, the ANOVA table indicates a significant two-factor interaction between crossover operator and mutation operator and a three-factor interaction involving the crossover operator, mutation probability, and selection rate.

**Table 5.9** Results for analysis of variance

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Model | 9 | 300.6 | 33.4004 | 2.26 | 0.025 |
| Linear | 4 | 90.57 | 22.6435 | 1.53 | 0.199 |
| Crossover Operator | 1 | 1.52 | 1.5203 | 0.10 | 0.749 |
| Mutation Probability | 1 | 0.73 | 0.7322 | 0.05 | 0.824 |
| Mutation Operator | 1 | 0 | 0 | 0 | 0.999 |
| Selection Rate | 1 | 88.32 | 88.3215 | 5.99 | 0.016 |
| 2-Way Interactions | 4 | 116.05 | 29.0118 | 1.97 | 0.106 |
| Crossover Operator*Mutation Probability | 1 | 70.04 | 70.0358 | 4.75 | 0.032 |
| Crossover Operator*Mutation Operator | 1 | 44.12 | 44.1189 | 2.99 | 0.087 |
| Crossover Operator*Selection Rate | 1 | 0 | 0.0021 | 0 | 0.991 |
| Mutation Probability*Selection Rate | 1 | 1.89 | 1.8905 | 0.13 | 0.721 |
| 3-Way Interactions | 1 | 93.98 | 93.9826 | 6.37 | 0.013 |
| Crossover Operator*Mutation Probability*Selection Rate | 1 | 93.98 | 93.9826 | 6.37 | 0.013 |
| Error | 90 | 1,327.82 | 14.7536 | | |
| Curvature | 1 | 18.14 | 18.1417 | 1.23 | 0.270 |
| Lack-of-Fit | 9 | 124.44 | 13.8269 | 0.93 | 0.501 |
| Pure Error | 80 | 1185.24 | 14.8154 | | |
| Total | 99 | 1628.42 | | | |

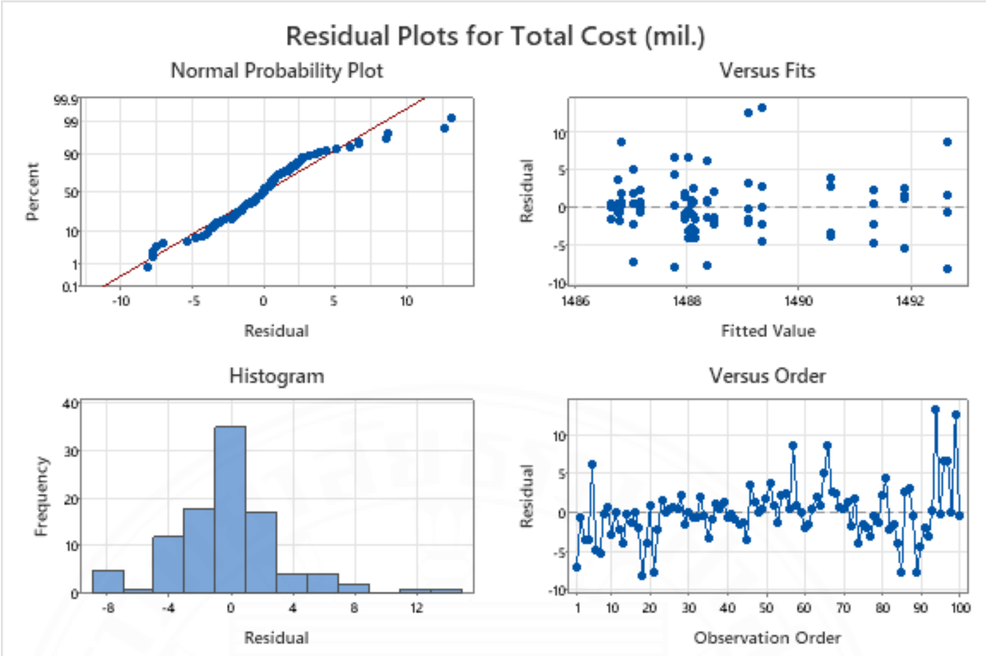



**Figure 5.6** Residual plots for response

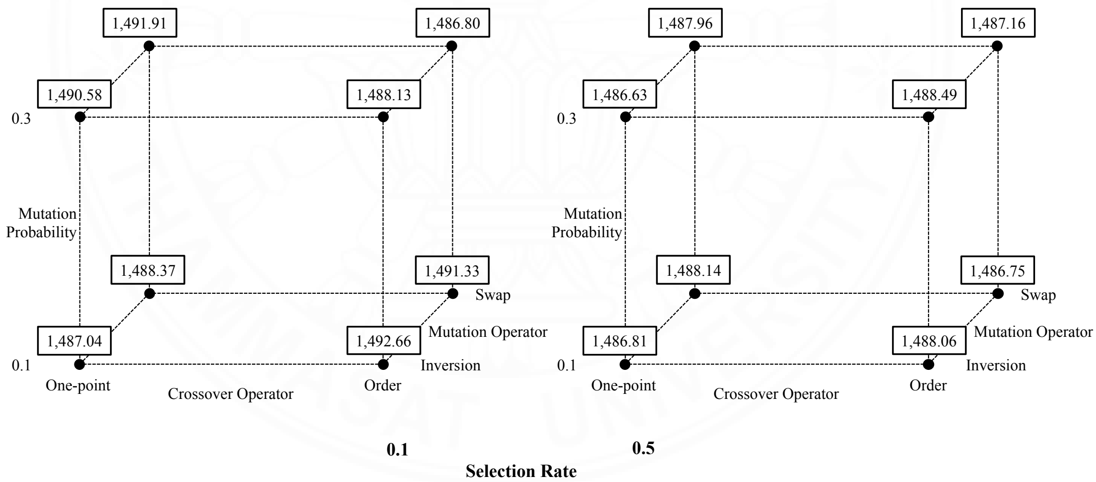



**Figure 5.7** Cube plot (fitted means) for response

To visually illustrate the effects of the three-factor interaction, a cube plot is presented in Figure 5.7. This cube plot effectively showcases the interaction among the crossover operator, mutation probability, and selection rate, helping us observe their combined impact on the GA's performance.

Remarkably, the cube plot identifies the optimal configuration for the GA as utilizing the one-point crossover operator, inversion mutation, a mutation probability of 0.3, and a selection rate of 0.5. This optimal configuration is further supported by the response optimizer, which is outlined in Table 5.10. It should be noted that multiple settings are given in the table, collectively yielding an expected total cost of approximately 1,487 million Baht.

**Table 5.10** Response optimizer

| Solution | Crossover Operator | Mutation Probability | Mutation Operator | Selection Rate | Response Fit | Composite Desirability |
|---|---|---|---|---|---|---|
| 1 | One-point | 0.300 | Inversion | 0.500 | 1,486.63 | 0.706484 |
| 2 | One-point | 0.296 | Inversion | 0.500 | 1,486.63 | 0.706335 |
| 3 | Order | 0.100 | Swap | 0.500 | 1,486.75 | 0.701219 |
| 4 | Order | 0.300 | Swap | 0.100 | 1,486.80 | 0.699050 |
| 5 | Order | 0.295 | Swap | 0.216 | 1,486.98 | 0.690708 |

The same GA hyperparameter tuning process is employed for optimizing the single-sourcing strategy. The most effective configuration involves utilizing order crossover, paired with an inversion mutation probability set at 0.3 of the probability, and a selection rate of 0.5. Details of the analysis are provided in Appendix D.

### 5.3.3 GA large-scale instance result

Once the parameters are optimized to their most effective values, the system's performance is assessed by solving the extended problem for both multi-sourcing and single-sourcing strategies. This iterative testing ensures the reliability and robustness of the GA's performance. The outcomes of this evaluation for both the multi-sourcing and single-sourcing strategies are depicted in Figure 5.8 and Figure 5.9.

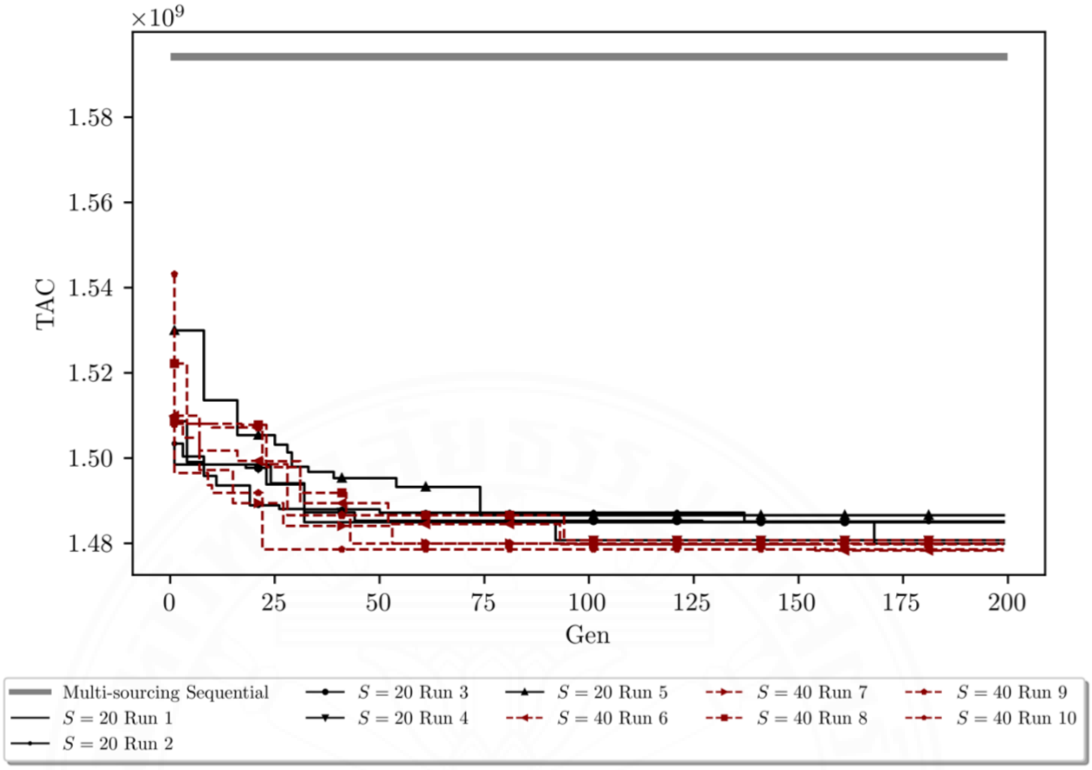



**Figure 5.8** GA result - multi-sourcing strategy – extended planning horizon

The proposed GA's performance is evaluated against outcomes derived from three different approaches: optimal solutions from the sequential approach for both multi-sourcing and single-sourcing strategies and the best solution found via the integrated approach for the single-sourcing strategy. The best-found solution obtained through the integrated approach, as shown in Figure 5.9, emerges after a 72-hour run and exhibits a 22.26% gap from the lower bound. It is noteworthy that, due to the intricate nature of the multi-sourcing strategy, the mathematical model in linear form could not be devised for this strategy under the integrated approach. Consequently, the integrated approach is solely applicable to the single-sourcing strategy.

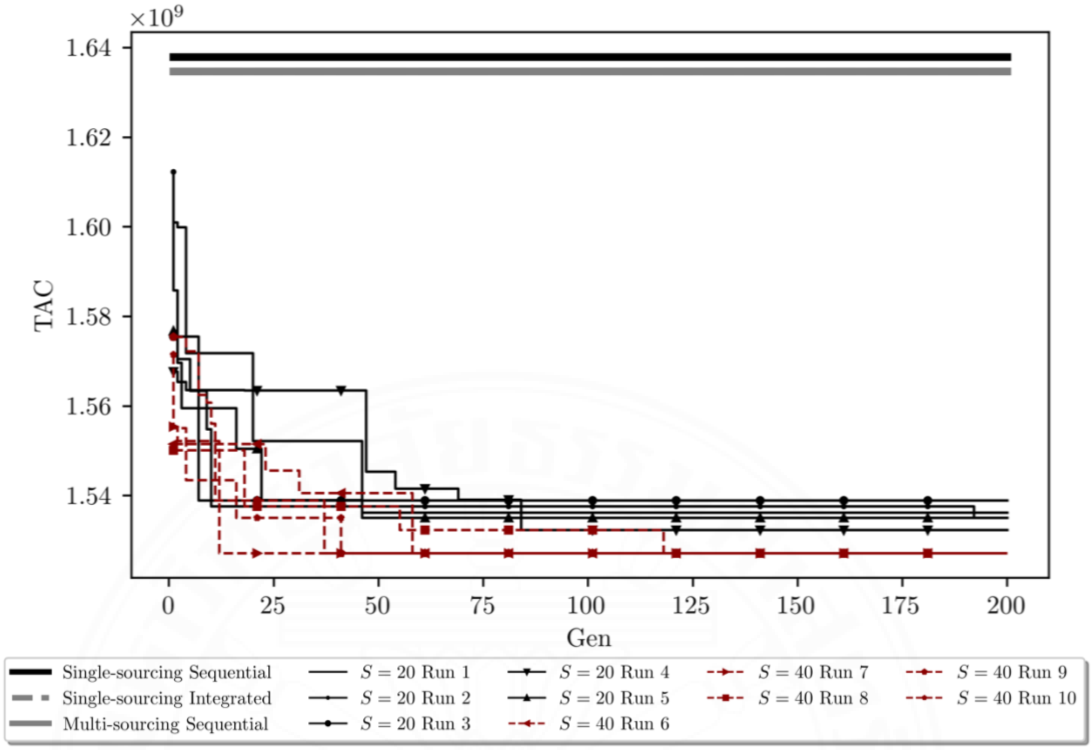



**Figure 5.9** GA result - single-sourcing strategy – extended planning horizon

It is demonstrated that as the population size increases, solutions improve, with this trend being more noticeable in the single-sourcing strategy. While improvements are also observable in the multi-sourcing strategy, they are comparatively more modest. These observations underscore the consistent performance of the proposed GA as the planning horizon extends from three periods to ten periods. This implies that the GA can yield proficient solutions for problems of medium to large scale. In addition, while CPLEX attains a solution with a 22.26% gap in 259,200 seconds, the GA is possible to achieve superior solutions in less computational time, i.e., with the population size of 20, 200 generations take approximately 34,000 seconds and 31,000 seconds for GAs of single-sourcing strategy and multi-sourcing strategy, respectively. The computational time doubles when the population size is increased to 40.

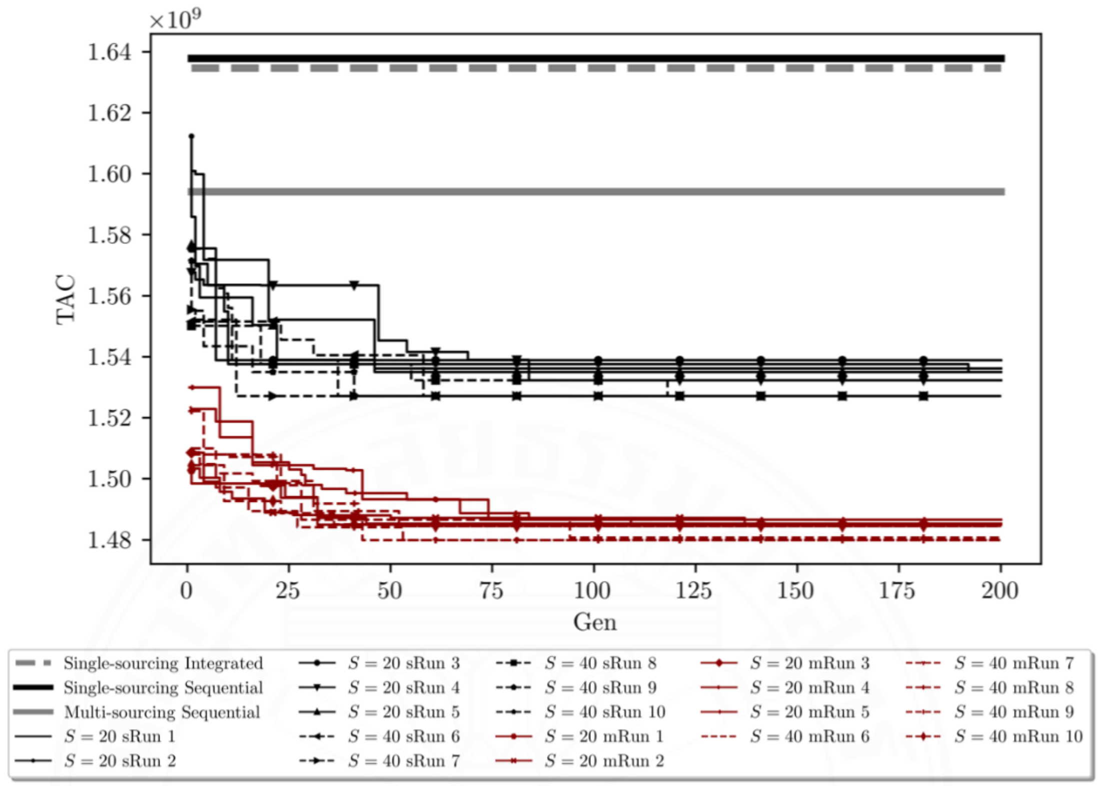



**Figure 5.10** GA result – both strategies – extended planning horizon

Additionally, Figure 5.10 underscores that, for identical problem instances, the multi-sourcing strategy (black lines in the graph) outperforms its single-sourcing counterpart (dark-red lines) in terms of total cost. This is due to the flexibility of the multi-sourcing strategy that offers diverse options for configuring the network configuration to lower the overall cost. Conversely, the single-sourcing strategy features a more confined network configuration, which lessens the potential alternatives for positioning SS. This underscores a trade-off between the simplicity of managing a single supply source with additional expenses and the intricacy of multi-sourcing management with lower costs.

# CHAPTER 6
# CONCLUSION AND FUTURE DIRECTIONS

## 6.1 Conclusion

This study introduces two innovative approaches that tackle the integration of ND and IP problems. The research framework addresses the complexities of multi-period, multi-echelon supply networks, incorporating bill of materials for diverse products and cost fluctuation across the planning horizon. Given specific cost parameters and final downstream facility demands, the demand uncertainty necessitates the establishment of SS at certain nodes to fulfill demand at a defined CSL. The advantages of these approaches are demonstrated through a numerical study applied to a chair manufacturing network.

A MILP model that integrates both production ND and IP concerns, specifically for the single-sourcing strategy, is formulated in Chapter 3. This MILP model not only determines an optimal network structure but simultaneously identifies the most suitable SS quantities and positions, resulting in a reduction in the total cost. A thorough comparison of the two approaches, including sequential and integrated approaches, reveals noteworthy cost benefits. The integrated approach consistently achieves cost savings between 1.7% and 3.7% compared to the conventional sequential approach, while CSLs range from 99% to 99.9%. Additionally, this integrated model maintains consistently high CSLs, indicating its ability to balance inventory holding and shortage costs while ensuring a high level of customer satisfaction.

In-depth sensitivity analysis is conducted to analyze the impact of distinct parameters – namely, service time and coefficient of variation – on the SC network structure and SS positions. The findings demonstrate the intricate interaction between these factors and costs. Specifically, higher demand variability, as indicated by an elevated $CV$, directly influences a rise in total SS costs, which consequently contributes to an overall escalation in the total cost. Additionally, increased committed service time correlates with a decline in total SS amount, leading to a corresponding decrease in total costs. Moreover, the selection of an optimal CSL is closely related to changes in SS costs, with higher CSLs favored when total SS costs decline and vice versa. This correlation between IP and CSL choice underscores the fundamental nature of the IP problem.

Chapter 4, on the other hand, introduces a GA to tackle the integration of ND and IP within the broader SC context, particularly focusing on large-scale scenarios. This versatile GA accommodates various sourcing strategies, including both single-sourcing and multi-sourcing scenarios. The proposed GA is developed to determine network structures, followed by optimizing flows via a minimum-cost flow problem and determining SS positioning through a MILP model. The GA's effectiveness is demonstrated through experiments conducted on a medium-scale problem, showing its capacity to achieve optimal solutions within reasonable computational time.

For large-scale challenges, a parameter tuning procedure is undertaken via a full factorial design. Four key parameters, including crossover operator, mutation probability, mutation operator, and selection rate, are considered. Using the optimal parameter setting, the GA is deployed on an extended planning horizon large-scale problem. The outcomes reveal the GA's superiority, surpassing optimal results derived from the sequential approach by 7.72% for the multi-sourcing strategy. Furthermore, the GA outperforms the best-found solution derived from the integrated approach by 7.04% for the single-sourcing problem.

In terms of total cost, the multi-sourcing strategy outperforms the single-sourcing strategy. However, the decision between these strategies necessitates a careful balance. Organizations need to weigh the advantages of simplicity against the need for resilience and cost optimization. Assessing factors such as supply chain robustness, cost-effectiveness, and risk mitigation becomes important in determining the most suitable sourcing strategy for a particular context or scenario within a supply chain. Specifically, a single supply source simplifies logistics, communication, and coordination activities. However, this simplicity often comes with additional expenses, primarily due to higher dependence on a singular source, potentially leading to increased vulnerability to supply disruptions. On the other hand, multi-sourcing strategies introduce complexity by managing multiple suppliers. Despite the intricacies involved, this approach significantly enhances supply chain resilience. By spreading the sourcing across various suppliers or locations, it mitigates the impact of potential disruptions. This diversified approach tends to result in lower costs in the long run, as it reduces the risk associated with relying solely on a single source.

## 6.2 Future directions

This study builds upon the foundations of Puga et al. (2019) and You and Grossmann (2008), extending their contributions by incorporating various enhancements. Our extensions encompass integrating multiple products with BOM, accommodating more diverse customer classes, expanding the supply chain network to a three-stage structure, introducing safety stocks for raw materials, and penalties for unmet demands through shortage cost of safety stock.

There are several directions for extending our research.

- Further extension could involve considering multiple modes of transportation and enabling direct shipments from one site to another.
- It is also imperative to incorporate uncertainty into the model's parameters, which could make it more realistic and robust. Stochastic optimization techniques could be employed to address demand and cost uncertainties, providing more reliable solutions.
- Due to the intricate nature of the problem, certain aspects from previous research, such as demand correlation and the selection between regular and express services, have not been thoroughly examined in this study. Future studies should aim to integrate these aspects. The incorporation of demand correlation significantly impacts safety stock management and can notably enhance decision-making concerning location-based inventory challenges.
- Involving the integration of risk pooling strategies into the design of responsive supply chains. This integration holds promise for further reductions in overall safety stock levels. However, it is important to note that such an extension may lead to a highly complex model. The computational challenges associated with this approach may make it intractable, particularly when dealing with intricate, multi-sourcing supply chain networks, even in small-scale instances.
- Extending the scope of the study to include closed-loop SCs, which involve product returns and recycling processes, along with adding environmental

considerations to the model. This could provide insights into sustainable SC design.

- An interdisciplinary approach holds immense potential to elevate the depth of our research. Integrating insights from diverse fields, such as other supply chain management's characteristics (considering product deterioration rate in agriculture, consumer, radio-active materials industry, etc.), economics (by examining pricing strategies, market behaviors, and the impact of supply chain decisions on economic systems), and environmental studies (by considering eco-friendly practices, reducing carbon footprints) can offer invaluable perspectives that may unlock novel dimensions of the integrated problem.
- Exploring alternative metaheuristics techniques and conducting a comparative analysis can provide valuable insights and enhance the robustness of the research.

# REFERENCES


Alshamsi, A., & Diabat, A. (2018). Large-scale reverse supply chain network design: An accelerated Benders decomposition algorithm. *Computers & Industrial Engineering, 124*, 545-559. doi: https://doi.org/10.1016/j.cie.2018.05.057

Altiparmak, F., Gen, M., Lin, L., & Karaoglan, I. (2009). A steady-state genetic algorithm for multi-product supply chain network design. *Computers & Industrial Engineering, 56*(2), 521-537. doi: https://doi.org/10.1016/j.cie.2007.05.012

Alumur, S. A., Nickel, S., Saldanha-da-Gama, F., & Verter, V. (2012). Multi-period reverse logistics network design. *European Journal of Operational Research, 220*(1), 67-68. doi: https://doi.org/10.1016/j.ejor.2011.12.045

Amin, S.H., & Zhang, G. (2013). A multi-objective facility location model for closed-loop supply chain network under uncertain demand and return. *Applied Mathematical Modelling, 37*, 4165–4176. https://doi.org/10.1016/j.apm.2012.09.039

Aouam, T., & Kumar, K. (2019). On the effect of overtime and subcontracting on supply chain safety stocks. *Omega*, *89*, 1-20.

Aouam, T., Ghadimi, F., & Vanhoucke, M. (2021). Finite inventory budgets in production capacity and safety stock placement under the guaranteed service approach. *Computers & Operations Research*, *131*, 105266.

Atamtürk, A., Berenguer, G., & Shen, Z.-J. (Max). (2012). A Conic Integer Programming Approach to Stochastic Joint Location-Inventory Problems. *Operations Research, 60*(2), 366–381. doi:10.1287/opre.1110.1037

Chen, H., & Li, P. (2015). Optimization of (R, Q) policies for serial inventory systems using the guaranteed service approach. *Computers & Industrial Engineering*, *80*, 261-273.

Clark, A. J., & Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. *Management science, 6*(4), 475-490.

Cortinhal, M. J., Lopes, M. J., & Melo, M. T. (2019). A multi-stage supply chain network design problem with in-house production and partial product outsourcing. *Applied Mathematical Modelling, 70*, 572-594. doi: https://doi.org/10.1016/j.apm.2019.01.046

Cui, Y. Y., Guan, Z., Zhang, L., Zhang, F., & Mirza, J. (2017). Close loop supply chain network problem with uncertainty in demand and returned products: Genetic artificial bee colony algorithm approach. *Journal of Cleaner Production, 162*, 717-742. https://doi.org/10.1016/j.jclepro.2017.06.079

Darmawan, A., Wong, H., & Thorstenson, A. (2021). Supply chain network design with coordinated inventory control. *Transportation Research Part E: Logistics and Transportation Review, 145*, 102168. doi: https://doi.org/10.1016/j.tre.2020.102168

De Kok, T., Grob, C., Laumanns, M., Minner, S., Rambau, J., & Schade, K. (2018). A typology and literature review on stochastic multi-echelon inventory models. *European Journal of Operational Research, 269*(3), 955-983.

Diabat, A., & Jebali, A. (2021). Multi-product and multi-period closed loop supply chain network design under take-back legislation. *International Journal of Production Economics, 231*, 107879. doi: https://doi.org/10.1016/j.ijpe.2020.107879

Durmaz, Y. G., & Bilgen, B. (2020). Multi-objective optimization of sustainable biomass supply chain network design. *Applied Energy, 272,* 115259. doi: https://doi.org/10.1016/j.apenergy.2020.115259

El-Beltagy, M. A., & Keane, A. J. (1999). A comparison of various optimization algorithms on a multilevel problem. *Engineering Applications of Artificial Intelligence*, *12*(5), 639-654. doi: https://doi.org/10.1016/S0952-1976(99)00033-0

Erbatur, F., Hasançebi, O. Tütüncü, I., & Kılıç, H. (2000). Optimal design of planar and space structures with genetic algorithms. *Computers & Structures, 75*(2), 209-224. doi: https://doi.org/10.1016/S0045-7949(99)00084-X

Eruguz, A. S., Sahin, E., Jemai, Z., & Dallery, Y. (2016). A comprehensive survey of guaranteed-service models for multi-echelon inventory optimization. *International Journal of Production Economics*, *172*, 110-125.

Fathollahi-Fard, A. M., Ahmadi, A., & Al-e-Hashem S.M.J. M. (2020). Sustainable closed-loop supply chain network for an integrated water supply and wastewater collection system under uncertainty. *Journal of Environmental Management, 275,* 111277. doi: https://doi.org/10.1016/j.jenvman.2020.111277

Funaki, K. (2012). Strategic safety stock placement in supply chain design with due-date based demand. *International Journal of Production Economics*, *135*(1), 4-13.

Gen, M., & Cheng, R. (2000). Genetic Algorithm and Engineering Optimization. *New York: John Wily and Sons*.

Gen, M., Altiparmak, F. & Lin, L. (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectrum*, 28, 337–354. doi: https://doi.org/10.1007/s00291-005-0029-9

Gen, M., Lin, L., Yun, Y., & Inoue, H. (2018). Recent advances in hybrid priority-based genetic algorithms for logistics and SCM network design. *Computers & Industrial Engineering, 125*, 394-412. doi: https://doi.org/10.1016/j.cie.2018.08.02556

Ghahremani-Nahr, J., Kian, R., & Sabet, E. (2019). A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm. *Expert Systems with Applications, 116*, 454-471. doi: https://doi.org/10.1016/j.eswa.2018.09.027

Ghadimi, F., Aouam, T., & Vanhoucke, M. (2020). Optimizing production capacity and safety stocks in general acyclic supply chains. *Computers & Operations Research*, *120*, 104938.

Gholizadeh, H., & Fazlollahtabar, H. (2020). Robust optimization and modified genetic algorithm for a closed loop green supply chain under uncertainty: Case study in melting industry. *Computers & Industrial Engineering, 147*, 106653. doi: https://doi.org/10.1016/j.cie.2020.106653

Gholizadeh, H., Goh, M., Fazlollahtabar, H., & Mamashli, Z. (2022). Modelling uncertainty in sustainable-green integrated reverse logistics network using metaheuristics optimization. *Computers & Industrial Engineering, 163*, 107828. doi: https://doi.org/10.1016/j.cie.2021.107828

Grahl, J., Minner, S., & Dittmar, D. (2016). Meta-heuristics for placing strategic safety stock in multi-echelon inventory with differentiated service times. *Annals of Operations Research*, 242, 489-504. https://doi.org/10.1007/s10479-014-1635-1

Graves, S. C. (1988). Safety Stocks in Manufacturing Systems. *Journal of Manufacturing and Operations Management, 1*, 67-101.

Graves, S. C., & Willems, S. P. (2000). Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management*, *2*(1), 68-83.

Graves, S. C., & Willems, S. P. (2005). Optimizing the Supply Chain Configuration for New Products. *Management Science, 51*(8), 1165-1180.

Graves, S. C., & Willems, S. P. (2008). Strategic inventory placement in supply chains: Nonstationary demand. *Manufacturing & Service Operations Management*, *10*(2), 278-287.

Graves, S. C., & Schoenmeyr, T. (2016). Strategic safety-stock placement in supply chains with capacity constraints. *Manufacturing & Service Operations Management*, *18*(3), 445-460.

Gutierrez, A., Dieulle, L., Labadie, N., & Velasco, N. (2018). A hybrid metaheuristic algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research, 99*, 135-147. doi: https://doi.org/10.1016/j.cor.2018.06.012

Hasani, A., Mokhtari, H., & Fattahi, M. (2021). A multi-objective optimization approach for green and resilient supply chain network design: A real-life case study. *Journal of Cleaner Production, 278*, 123199. doi: https://doi.org/10.1016/j.jclepro.2020.123199

Hax, A.C. (1973). Integration of Strategic and Tactical Planning in the Aluminum Industry. *Massachusetts Institute of Technology, Operations Research Center*.

Holland, J. (1975). Adaptation in natural and artificial systems. *University of Michigan press, Ann Arbor.*

Humair, S., & Willems, S. P. (2011). TECHNICAL NOTE---Optimizing Strategic Safety Stock Placement in General Acyclic Networks. *Operations Research, 59*(3), 781-787.

Jiang, D., Li, H., Yang, T., & Li, D. (2016). Genetic algorithm for inventory positioning problem with general acyclic supply chain networks. *European journal of industrial engineering*, *10*(3), 367-384.

John, S. T., Sridharan, R., Kumar, P. N., & Krishnamoorthy, M. (2018). Multi-period reverse logistics network design for used refrigerators. *Applied Mathematical Modelling, 54,* 311-331. https://doi.org/10.1016/j.apm.2017.09.053

Jung, J. Y., Blau, G., Pekny, J. F., Reklaitis, G. V., & Eversdyk, D. (2008). Integrated safety stock management for multi-stage supply chains under production capacity constraints. *Computers & Chemical Engineering*, *32*(11), 2570-2581.

Kaminsky, P., & Kaya, O. (2008). Inventory positioning, scheduling and lead-time quotation in supply chains. *International Journal of Production Economics*, *114*(1), 276-293.

Kimball, G. (1988). General principles of inventory control. *Journal of Manufacturing and Operations Management, 1*, 119–130.

Klosterhalfen, S. T., Minner, S., & Willems, S. P. (2014). Strategic safety stock placement in supply networks with static dual supply. *Manufacturing & Service Operations Management*, *16*(2), 204-219. https://doi.org/10.1287/msom.2013.0472

Lee, J-E., Gen, M., & Rhee, K-G. (2009). Network model and optimization of reverse logistics by hybrid genetic algorithm. *Computers & Industrial Engineering, 56*(3), 951-964. doi: https://doi.org/10.1016/j.cie.2008.09.021

Li, P., & Chen, H. (2012). Optimal $(R, nQ)$ Policies for Serial Inventory Systems with Guaranteed Service. In: Klatte, D., Lüthi, HJ., Schmedders, K. (eds) Operations Research Proceedings 2011. *Operations Research Proceedings*. Springer, Berlin, Heidelberg. pp. 439-444.

Li, P., Chen, H., & Che, A. (2013). Optimal batch ordering policies for assembly systems with guaranteed service. *International Journal of Production Research*, *51*(20), 6275-6293.

Lin, L., Gen, M., & Wang, X. (2009). Integrated multistage logistics network design by using hybrid evolutionary algorithm. *Computers & Industrial Engineering, 56*(3), 854-873. doi: https://doi.org/10.1016/j.cie.2008.09.037

Lofti, M. M., & Tavakkoli-Moghaddam, R. (2013). A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. *Applied Soft Computing, 13*(5), 2711-2726. doi: https://doi.org/10.1016/j.asoc.2012.11.016

Moncayo-Martínez, L. A., & Zhang, D. Z. (2013). Optimising safety stock placement and lead time in an assembly supply chain using bi-objective MAX–MIN ant system. *International Journal of Production Economics*, *145*(1), 18-28.

Nezamoddini, N., Gholami, A., & Aqlan, F. (2020). A risk-based optimization framework for integrated supply chains using genetic algorithm and artificial neural networks. *International Journal of Production Economics, 225*, 107569. https://doi.org/10.1016/j.ijpe.2019.107569

Paithankar, A., & Chatterjee, S. (2019). Open pit mine production schedule optimization using a hybrid of maximum-flow and genetic algorithms. *Applied Soft Computing, 81*, 105507. doi: https://doi.org/10.1016/j.asoc.2019.105507

Pazhani, S., Mendoza, A., Nambirajan, R., Narendran, T. T., Ganesh, K., & Olivares-Benitez, E. (2021). Multi-period multi-product closed loop supply chain network design: A relaxation approach. *Computers & Industrial Engineering, 155*, 107191. doi: https://doi.org/10.1016/j.cie.2021.107191

Puga, M. S., Minner, S., & Tancrez, J-S. (2019). Two-stage supply chain design with safety stock placement decisions. *International Journal of Production Economics, 209*, 183-193. doi: https://doi.org/10.1016/j.ijpe.2018.05.018

Rahmaniani, R., & Ghaderi, A. (2013). A combined facility location and network design problem with multi-type of capacitated links. *Applied Mathematical Modelling, 37*, 6400–6414. doi: https://doi.org/10.1016/j.apm.2013.01.001

Robles, J. O., Azzaro-Pantel, C., & Aguilar-Lasserre, A. (2020). Optimization of a hydrogen supply chain network design under demand uncertainty by multi-objective genetic algorithms. *Computers and Chemical Engineering, 140*, 106853. doi: https://doi.org/10.1016/j.compchemeng.2020.106853

Rodriguez, M. A., Vecchietti, A. R., Harjunkoski, I., & Grossmann, I. E. (2014). Optimal supply chain design and management over a multi-period horizon under demand uncertainty. Part I: MINLP and MILP models. *Computers & Chemical Engineering, 62*, 194-210. doi: https://doi.org/10.1016/j.compchemeng.2013.10.007

Rohaninejad, M., Sahraeian, R., & Tavakkoli-Moghaddam, R. (2018). Multi-echelon supply chain design considering unreliable facilities with facility hardening possibility. *Applied Mathematical Modelling, 62*, 321–337. doi: https://doi.org/10.1016/j.apm.2018.06.001

Savadkoohi, E., Mousazadeh, M., & Torabi, S. A. (2018). A possibilistic Location-Inventory model for multi-period perishable pharmaceutical supply chain network design. *Chemical Engineering Research and Design, 138*, 490-505. doi: https://doi.org/10.1016/j.cherd.2018.09.008

Seshadri, S., Chatterjee, K., & Lilien, G. L. (1991). Multiple Source Procurement Competitions. *Marketing Science, 10*(3), 246–263. doi: http://www.jstor.org/stable/183944

Shenas, N.Y., Jahromi, A.E., & Yazdi, M.M. (2009). An Efficient Procedure for Computing an Optimal $(R, Q)$ Policy in Continuous Review Systems with Poisson Demands and Constant Lead Time. *Transaction E: Industrial Engineering, 16*(2), 128-137.

Shi, J., Liu, Z., Tang, L., & Xiong, J. (2017). Multi-objective optimization for a closed-loop network design problem using an improved genetic algorithm. *Applied Mathematical Modelling, 45*, 14-30. doi: https://doi.org/10.1016/j.apm.2016.11.004

Shoja, A., Molla-Alizadeh-Zavardehi, S., & Niroomand, S. (2019). Adaptive meta-heuristic algorithms for flexible supply chain network design problem with different delivery modes. *Computers & Industrial Engineering, 138*, 106107. doi: https://doi.org/10.1016/j.cie.2019.106107

Shoja, A., Molla-Alizadeh-Zavardehi, S., & Niroomand, S. (2020). Hybrid adaptive simplified human learning optimization algorithms for supply chain network design problem with possibility of direct shipment. *Applied Soft Computing, 96*, 106594. doi: https://doi.org/10.1016/j.asoc.2020.106594

Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2008). Designing and managing the supply chain: Concepts, strategies, and cases (3rd edition). *United-States: McGraw-Hill*.

Simchi-Levi, D., & Zhao, Y. (2012). Performance evaluation of stochastic multi-echelon inventory systems: A survey. *Advances in Operations Research, 2012*, 1–34.

Simpson Jr, K. F. (1958). In-process inventories. *Operations Research*, *6*(6), 863-873.

Smith. S. (1978). A two-phase method for timber supply analysis. *Proceedings of IUFRO Operational forest management planning methods, Bucharest, Romania*, 76-81

Tari, F. G., & Hashemi, Z. (2016). A priority based genetic algorithm for nonlinear transportation costs problems. *Computers & Industrial Engineering, 96*, 86-95. doi: https://doi.org/10.1016/j.cie.2016.03.010

Taxakis, K., & Papadopoulos, C. (2016). A design model and a production–distribution and inventory planning model in multi-product supply chain networks. *International Journal of Production Research, 54*(21), 6436-6457. doi: 10.1080/00207543.2016.1158882

Thompson, J. D. (2016). Statistical Alignment Approaches. Statistics for Bioinformatics. *Elsevier*, 43-51. https://doi.org/10.1016/B978-1-78548-216-8.50004-X

Treleven, M., & Schweikhart, S. B. (1988). A risk/benefit analysis of sourcing strategies: Single vs. multiple sourcing. *Journal of Operations Management, 7*(4), 93–114. doi: https://doi.org/10.1016/0272-6963(81)90007-3

Wang, Q. (2011). Control Policies for Multi-echelon Inventory Systems with Stochastic Demand. In Supply Chain Coordination under Uncertainty. *International Handbooks on Information Systems. Berlin, Heidelberg: Springer*, pp. 83–108.

Weintraub, A.P., & Navon. D. (1976). A forest management planning model integrating silvicultural and transportation activities. *Management Science, 22*(12), 1299-1309. doi: https://doi.org/10.1287/mnsc.22.12.1299

Weintraub, A.P., & Cholaky, A. (1991). A hierarchical approach to forest planning. *Forest Science, 37* (2), 439-460.

Wong, Y. Y., Lee, K. H., Leung, K. S., & Ho, C.-W. (2003). A novel approach in parameter adaptation and diversity maintenance for genetic algorithms. *Soft Computing*, *7*, 506–515. doi: https://doi.org/10.1007/s00500-002-0235-1

You, F., & Grossmann, I. E. (2008). Design of responsive supply chains under demand uncertainty. *Computers & Chemical Engineering, 32*(12), 3090-3111. doi: https://doi.org/10.1016/j.compchemeng.2008.05.004

You, F., & Grossmann, I. E. (2010). Integrated multi-echelon supply chain design with inventories under uncertainty: MINLP models, computational strategies. *AIChE J., 56*, 419-440. doi: https://doi.org/10.1002/aic.12010

Zadeh, A. S., Sahraeian, R., & Homayouni, S. M. (2014). A dynamic multi-commodity inventory and facility location problem in steel supply chain network design. *Int*

*J Adv Manuf Technol, 70,* 1267–1282. doi: https://doi.org/10.1007/s00170-013-5358-2

Zeballos, L. J., Méndez, C. A., Barbosa-Povoa, A. P., & Novais, A. Q. (2014). Multi-period design and planning of closed-loop supply chains with uncertain supply and demand. *Computers & Chemical Engineering, 66*, 151-164. doi: https://doi.org/10.1016/j.compchemeng.2014.02.027

# APPENDICES

# APPENDIX A
# THE ORIGINAL MIXED-INTEGER NONLINEAR PROGRAMMING MODEL OF THE INTEGRATED PROBLEM

When integrating the two problems, $\sigma_{a,j,k,t}$ and $\sigma_{c,i,j,t}$ are not known in advance. They become decision variables in the MINLP model.

$$\begin{aligned} Z_N = Z_1 &+ \sum_{c\in C}\sum_{i\in L_1}\sum_{j\in L_2}\sum_{t\in T}\mathcal{HC}_{c,i,t}\,\Phi^{-1}(s)\sigma_{c,i,j,t}\left(\sqrt{\mathcal{XC}_{c,i,j,t}}+\sqrt{\mathcal{YC}_{c,i,j,t}}\right)\\ &+\sum_{a\in A}\sum_{j\in L_2}\sum_{k\in K}\sum_{t\in T}\mathcal{HA}_{a,j,t}\Phi^{-1}(s)\sigma_{a,j,k,t}\left(\sqrt{\mathcal{XA}_{a,j,k,t}}+\sqrt{\mathcal{YA}_{a,j,k,t}}\right)\\ &+\sum_{p\in P}\sum_{k\in K}\sum_{t\in T}\mathcal{HP}_{p,k,t}\Phi^{-1}(s)\sigma_{p,k,t}\sqrt{\mathcal{XP}_{p,k,t}}\\ &+\sum_{c\in C}\sum_{i\in L_1}\sum_{j\in L_2}\sum_{t\in T}\mathcal{LC}_{c,i,t}\,\mathcal{L}(s)\sigma_{c,i,j,t}\left(\frac{365}{\mathcal{XC}_{c,i,j,t}}\sqrt{\mathcal{XC}_{c,i,j,t}}+\frac{365}{\mathcal{YC}_{c,i,j,t}}\sqrt{\mathcal{YC}_{c,i,j,t}}\right)\\ &+\sum_{a\in A}\sum_{j\in L_2}\sum_{k\in K}\sum_{t\in T}\mathcal{LA}_{a,j,t}\mathcal{L}(s)\sigma_{a,j,k,t}\left(\frac{365}{\mathcal{XA}_{a,j,k,t}}\sqrt{\mathcal{XA}_{a,j,k,t}}+\frac{365}{\mathcal{YA}_{a,j,k,t}}\sqrt{\mathcal{YA}_{a,j,k,t}}\right)\\ &+\sum_{p\in P}\sum_{k\in K}\sum_{t\in T}\mathcal{LP}_{p,k,t}\mathcal{L}(s)\sigma_{p,k,t}\frac{365}{\mathcal{XP}_{p,k,t}}\sqrt{\mathcal{XP}_{p,k,t}} \end{aligned} \tag{1}$$

$$Minimize\ Z_N$$

Subject to

Constraints (3.2)-(3.21), (3.43)-(3.47), (3.58), (3.69)-(3.73),

$$\sigma_{c,i,j,t} = CC_{c,i,j,t}\sqrt{\beta^2\alpha^2\sum_{k\in D}CA_{a,j,k,t}\sigma^2_{p,k,t}} \tag{2}$$
$$\forall (c,a)\in BOM_C,(a,p)\in BOMA\ i\in L_1, j\in L_2, t\in T$$

$$\sigma_{a,j,k,t} = \sqrt{\alpha^2\sum_{k\in D}CA_{a,j,k,t}\sigma^2_{p,k,t}} \tag{3}$$
$$\forall (a,p)\in BOMC, i\in L_1, j\in L_2, t\in T$$

Constraints (2) and (3) are to determine the SDs of arcs connecting $L_1$ with $L_2$ and connecting $L_2$ with $D$, respectively.

# APPENDIX B
# LONGEST PATH MODEL

This is the MILP model to find the longest path in the network. Solution to this problem is used the maximum index for the RT $\mathcal{M}$.

Because the IP model contains non-linear terms, to linearize $Z_2$, in a range from 0 to a predefined maximum value of the RT $m$, all potential values of $m$ are listed. The longest path from one of the most upstream component manufacturers to the last downstream manufacturers in the network is determined as the maximum value of $m$ by the following model.

**Decision variables:**

$XC_i$: the processing time of node $i \in L_1$ if it is selected;

$XA_j$: the processing time of node $j \in L_2$ if it is selected;

$XP_k$: the processing time of node $k \in K$ if it is selected;

$YC_{i,j}$: 1 if an arc from component manufacturer $i \in L_1$ to subassembly manufacturer $j \in L_2$ is selected, 0 otherwise;

$YA_{j,k}$: 1 if an arc from subassembly manufacturer $j \in L_2$ to final manufacturer $k \in D$ is selected, 0 otherwise;

$$Maximize\ LP = \sum_{i \in L_1} XC_i + \sum_{j \in L_2} XA_j + \sum_{k \in K} XP_k + \sum_{i \in L_1} \sum_{j \in L_2} \mathcal{OC}_{i,j} YC_{i,j} + \sum_{j \in L_2} \sum_{k \in K} \mathcal{OA}_{j,k} YA_{j,k} \quad (4)$$

Subject to

$$\sum_{j \in L_2} YC_{i,j} \leq 1 \ \forall i \in L_1 \quad (5)$$

$$\sum_{i \in L_1} YC_{i,j} - \sum_{k \in D} YA_{j,k} = 0 \ \forall j \in L_2 \quad (6)$$

$$\sum_{j \in L_2} YA_{j,k} \leq 1 \ \forall k \in K \quad (7)$$

$$\mathcal{PC}_i \sum_{j \in L_2} YC_{i,j} \geq XC_i \quad \forall i \in L_1 \tag{8}$$

$$\mathcal{PA}_j \sum_{k \in D} YA_{j,k} \geq XA_j \quad \forall j \in L_2 \tag{9}$$

$$\mathcal{PP}_k \sum_{j \in L_2} YA_{j,k} \geq XP_k \quad \forall k \in K \tag{10}$$

$$XC_i, XA_j, XP_k \geq 0, \quad \forall i \in L_1, j \in L_2, k \in K \tag{11}$$

$$YC_{i,j}, YA_{j,k} = \{0,1\}, \quad \forall i \in L_1, j \in L_2, k \in K \tag{12}$$

The objective function (4) maximizes the total number of periods on the longest path which consists of the transit time $\mathcal{OC}_{i,j}$, $\mathcal{OA}_{j,k}$ and processing time $\mathcal{PC}_i, \mathcal{PA}_j$, and $\mathcal{PP}_k$. Constraints (5) and (7) ensure that the longest path only contains one arc connecting $L_1$ with $L_2$ and one arc connecting $L_2$ with $K$. Constraint (6) ensure that the two selected arcs share the same node in $L_2$. Constraints (8) - (10) determine the processing time of a node on the longest path. Constraints (11) and (12) represent binary variables.

# APPENDIX C

# $2^4$ FULL FACTORIAL DESIGN WITH CENTER POINTS

**Table C.1** The multi-sourcing strategy

| Run Order | Center Point | Blocks | Crossover Operator | Mutation Probability | Mutation Operator | Selection Rate | Total Cost (Million BHT) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,480.00 |
| 2 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,492.15 |
| 3 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,487.20 |
| 4 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,484.56 |
| 5 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,494.51 |
| 6 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,486.60 |
| 7 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,486.60 |
| 8 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,486.65 |
| 9 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,487.56 |
| 10 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,485.37 |
| 11 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,486.77 |
| 12 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,486.29 |
| 13 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,484.13 |
| 14 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,486.64 |
| 15 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,486.60 |
| 16 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,487.27 |
| 17 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,485.01 |
| 18 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,484.56 |
| 19 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,486.77 |
| 20 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,488.99 |
| 21 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,480.79 |
| 22 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,489.15 |
| 23 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,493.62 |
| 24 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,486.77 |
| 25 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,487.20 |
| 26 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,488.72 |
| 27 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,487.20 |
| 28 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,490.69 |
| 29 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,486.64 |
| 30 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,486.77 |
| 31 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,487.27 |
| 32 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,486.60 |
| 33 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,488.99 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 34 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,492.21 |
| 35 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,487.27 |
| 36 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,487.20 |
| 37 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,489.51 |
| 38 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,491.91 |
| 39 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,493.39 |
| 40 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,486.29 |
| 41 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,486.60 |
| 42 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,487.20 |
| 43 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,485.11 |
| 44 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,487.27 |
| 45 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,484.56 |
| 46 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,490.42 |
| 47 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,489.39 |
| 48 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,487.20 |
| 49 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,487.58 |
| 50 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,494.42 |
| 51 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,494.51 |
| 52 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,488.99 |
| 53 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,487.20 |
| 54 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,493.62 |
| 55 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,494.42 |
| 56 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,487.20 |
| 57 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,495.54 |
| 58 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,489.13 |
| 59 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,486.77 |
| 60 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,486.64 |
| 61 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,486.65 |
| 62 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,487.27 |
| 63 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,490.00 |
| 64 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,488.01 |
| 65 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,492.21 |
| 66 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,501.30 |
| 67 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,493.32 |
| 68 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,490.69 |
| 69 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,489.13 |
| 70 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,491.91 |
| 71 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,493.22 |
| 72 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,485.11 |
| 73 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,488.72 |
| 74 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,484.13 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 75 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,485.11 |
| 76 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,486.60 |
| 77 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,485.01 |
| 78 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,486.29 |
| 79 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,486.60 |
| 80 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,489.51 |
| 81 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,492.16 |
| 82 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,487.27 |
| 83 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,487.63 |
| 84 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,484.13 |
| 85 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,480.00 |
| 86 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,492.08 |
| 87 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,492.35 |
| 88 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,487.52 |
| 89 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,480.00 |
| 90 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,485.01 |
| 91 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,487.27 |
| 92 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,485.01 |
| 93 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,488.01 |
| 94 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,502.59 |
| 95 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,488.99 |
| 96 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,494.77 |
| 97 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,494.51 |
| 98 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,489.39 |
| 99 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,501.79 |
| 100 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,487.58 |

**Table C.2** The single-sourcing strategy

| Run Order | Center Point | Blocks | Crossover Operator | Mutation Probability | Mutation Operator | Selection Rate | Total Cost (Million BHT) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,531.73 |
| 2 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,539.10 |
| 3 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,536.94 |
| 4 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,536.27 |
| 5 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,535.11 |
| 6 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,545.98 |
| 7 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,531.73 |
| 8 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,539.17 |
| 9 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,536.27 |
| 10 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,532.37 |
| 11 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,531.73 |
| 12 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,527.16 |
| 13 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,535.11 |
| 14 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,527.16 |
| 15 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,535.11 |
| 16 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,536.27 |
| 17 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,532.37 |
| 18 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,532.37 |
| 19 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,531.73 |
| 20 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,527.16 |
| 21 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,535.04 |
| 22 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,539.10 |
| 23 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,532.37 |
| 24 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,536.27 |
| 25 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,535.11 |
| 26 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,527.16 |
| 27 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,532.37 |
| 28 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,531.73 |
| 29 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,536.27 |
| 30 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,531.73 |
| 31 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,539.17 |
| 32 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,536.27 |
| 33 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,535.04 |
| 34 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,531.73 |
| 35 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,527.16 |
| 36 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,527.16 |
| 37 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,532.37 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 38 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,535.04 |
| 39 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,527.16 |
| 40 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,539.17 |
| 41 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,535.04 |
| 42 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,531.73 |
| 43 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,535.11 |
| 44 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,527.16 |
| 45 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,535.11 |
| 46 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,535.04 |
| 47 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,532.37 |
| 48 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,535.04 |
| 49 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,535.11 |
| 50 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,532.37 |
| 51 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,531.73 |
| 52 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,531.73 |
| 53 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,532.37 |
| 54 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,539.10 |
| 55 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,531.73 |
| 56 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,544.31 |
| 57 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,532.37 |
| 58 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,532.37 |
| 59 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,535.11 |
| 60 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,531.73 |
| 61 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,531.73 |
| 62 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,532.37 |
| 63 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,536.27 |
| 64 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,536.27 |
| 65 | 1 | 1 | 1-point | 0.1 | inversion | 0.1 | 1,539.10 |
| 66 | 1 | 1 | order | 0.1 | inversion | 0.1 | 1,536.27 |
| 67 | 1 | 1 | 1-point | 0.3 | inversion | 0.1 | 1,536.27 |
| 68 | 1 | 1 | order | 0.3 | inversion | 0.1 | 1,527.16 |
| 69 | 1 | 1 | 1-point | 0.1 | swap | 0.1 | 1,536.27 |
| 70 | 1 | 1 | order | 0.1 | swap | 0.1 | 1,535.11 |
| 71 | 1 | 1 | 1-point | 0.3 | swap | 0.1 | 1,527.16 |
| 72 | 1 | 1 | order | 0.3 | swap | 0.1 | 1,536.27 |
| 73 | 1 | 1 | 1-point | 0.1 | inversion | 0.5 | 1,535.04 |
| 74 | 1 | 1 | order | 0.1 | inversion | 0.5 | 1,531.73 |
| 75 | 1 | 1 | 1-point | 0.3 | inversion | 0.5 | 1,536.27 |
| 76 | 1 | 1 | order | 0.3 | inversion | 0.5 | 1,527.16 |
| 77 | 1 | 1 | 1-point | 0.1 | swap | 0.5 | 1,535.11 |
| 78 | 1 | 1 | order | 0.1 | swap | 0.5 | 1,527.16 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 79 | 1 | 1 | 1-point | 0.3 | swap | 0.5 | 1,535.04 |
| 80 | 1 | 1 | order | 0.3 | swap | 0.5 | 1,535.11 |
| 81 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,527.16 |
| 82 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,532.37 |
| 83 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,535.11 |
| 84 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,535.11 |
| 85 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,527.16 |
| 86 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,527.16 |
| 87 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,531.73 |
| 88 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,535.04 |
| 89 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,527.16 |
| 90 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,531.73 |
| 91 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,532.37 |
| 92 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,532.37 |
| 93 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,531.73 |
| 94 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,535.04 |
| 95 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,535.11 |
| 96 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,536.27 |
| 97 | 0 | 1 | 1-point | 0.2 | inversion | 0.3 | 1,532.37 |
| 98 | 0 | 1 | order | 0.2 | inversion | 0.3 | 1,539.10 |
| 99 | 0 | 1 | 1-point | 0.2 | swap | 0.3 | 1,535.11 |
| 100 | 0 | 1 | order | 0.2 | swap | 0.3 | 1,539.10 |

# APPENDIX D
# $2^4$ FACTORIAL DESIGN ANALYSIS RESULT FOR THE SINGLE-SOURCING STRATEGY

**Table D.1** Results for analysis of variance

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Model | 12 | 663.4 | 55.283 | 6.07 | 0.000 |
| Linear | 4 | 202.8 | 50.7 | 5.57 | 0.000 |
| Crossover Operator | 1 | 3.11 | 3.112 | 0.34 | 0.560 |
| Mutation Probability | 1 | 15.81 | 15.806 | 1.74 | 0.191 |
| Mutation Operator | 1 | 162.97 | 162.971 | 17.9 | 0.000 |
| Selection Rate | 1 | 20.91 | 20.91 | 2.3 | 0.133 |
| 2-Way Interactions | 6 | 368.11 | 61.352 | 6.74 | 0.000 |
| Crossover Operator*Mutation Probability | 1 | 6.45 | 6.452 | 0.71 | 0.402 |
| Crossover Operator*Mutation Operator | 1 | 95.96 | 95.962 | 10.54 | 0.002 |
| Crossover Operator*Selection Rate | 1 | 166.29 | 166.291 | 18.26 | 0.000 |
| Mutation Probability*Mutation Operator | 1 | 38.53 | 38.531 | 4.23 | 0.043 |
| Mutation Probability*Selection Rate | 1 | 58.65 | 58.653 | 6.44 | 0.013 |
| Mutation Operator*Selection Rate | 1 | 2.22 | 2.224 | 0.24 | 0.622 |
| 3-Way Interactions | 2 | 92.48 | 46.241 | 5.08 | 0.008 |
| Crossover Operator*Mutation Probability *Mutation Operator | 1 | 49.24 | 49.235 | 5.41 | 0.022 |
| Crossover Operator*Mutation Operator *Selection Rate | 1 | 43.25 | 43.247 | 4.75 | 0.032 |
| Error | 87 | 792.12 | 9.105 | | |
| Curvature | 1 | 10.42 | 10.42 | 1.15 | 0.287 |
| Lack-of-Fit | 6 | 119.14 | 19.857 | 2.4 | 0.035 |
| Pure Error | 80 | 662.56 | 8.282 | | |
| Total | 99 | 1455.52 | | | |

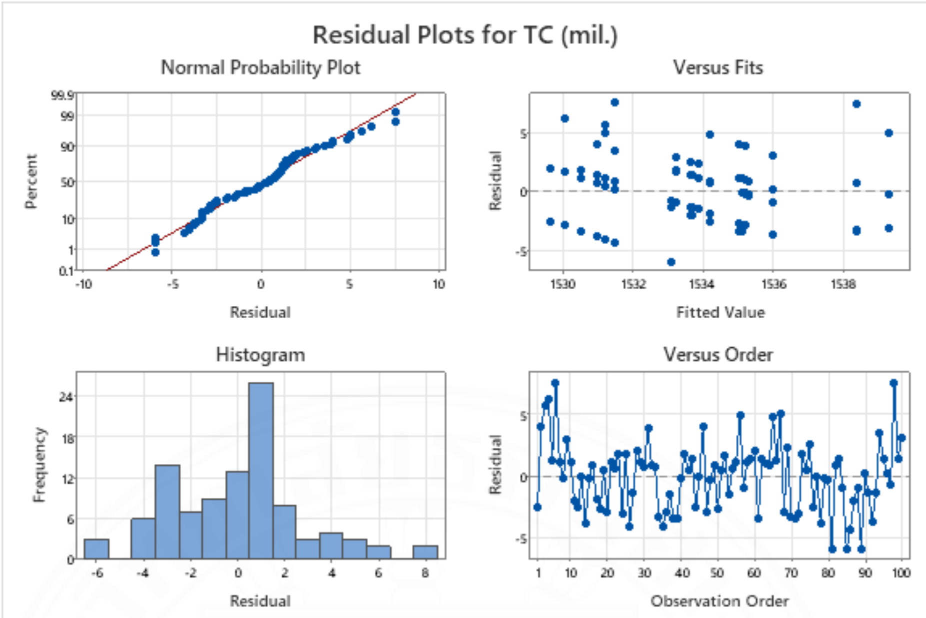



**Figure D.1** Residual plots for response

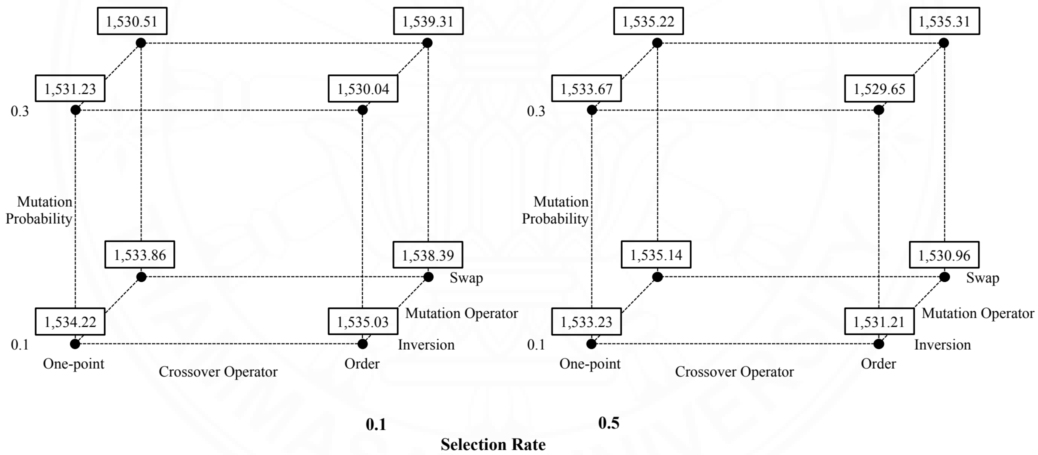



**Figure D.2** Cube plot (fitted means) for response

**Table D.2** Response optimizer

| Solution | Crossover Operator | Mutation Probability | Mutation Operator | Selection Rate | Response Fit | Composite Desirability |
|---|---|---|---|---|---|---|
| 1 | Order | 0.300 | Inversion | 0.500 | 1,529.65 | 0.867851 |
| 2 | Order | 0.300 | Inversion | 0.102 | 1,530.03 | 0.847281 |
| 3 | One-point | 0.300 | Swap | 0.100 | 1,530.51 | 0.822160 |
| 4 | One-point | 0.282 | Swap | 0.102 | 1,530.84 | 0.804670 |
| 5 | Order | 0.100 | Swap | 0.500 | 1,530.96 | 0.797877 |

# BIOGRAPHY

| | |
|---|---|
| Name | Nguyen Trong Tri Duc |
| Education | 2018: Bachelor of Engineering (Industrial Management)<br>College of Engineering Technology<br>Can Tho University<br>2020: Master of Engineering (Logistics and Supply Chain Systems Engineering)<br>Sirindhorn International Institute of Technology<br>Thammasat University |

Publications

Tham, T.T., Duc, N.T.T., Dung, T.T.M., & Nguyen, H.-P. (2020). An integrated approach of ISM and fuzzy TOPSIS for supplier selection. *International Journal of Procurement Management*, *13*(5), 701–735.

Duc, N. T. T., Tai, P. D., & Buddhakulsomsiri, J. (2020). Approximating Measures of Performance of a Periodic Review Inventory System by Using Markov Chain. *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)* (pp. 543-547). Bangkok, Thailand.

Duc, N. T. T., Tai, P. D., & Buddhakulsomsiri, J. (2022). A Markovian approach to modeling a periodic order-up-to-level policy under stochastic discrete demand and lead time with lost sales. *International Transactions in Operational Research*, *29*(2), 1132-1158. doi: https://doi.org/10.1111/itor.13042

Duc, N. T. T., Tai, P. D., & Buddhakulsomsiri, J. (2022). Inventory Positioning in Supply Chain Network: A Service-Oriented Approach. *IEEE Access*, *10*, 92986-93002. doi: 10.1109/ACCESS.2022.3202906.

Duc, N. T. T., Buddhakulsomsiri, J., & Tai, P. D. (2023). Rank-Based Encoder and Decoder for a Genetic Algorithm: An Application to Network Design Problem.

*2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE)* (pp. 716-721). Yunlin, Taiwan. doi: 10.1109/ECICE59523.2023.10383086.