Titre: Title:	Sustainable Physical Internet with Uncertain Flow-Dependent Travel Times: Modeling and Solution Approach
Auteur: Author:	Marjan Sadeghi
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Sadeghi, M. (2024). Sustainable Physical Internet with Uncertain Flow-Dependent Travel Times: Modeling and Solution Approach [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/62052/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/62052/
Directeurs de recherche: Advisors:	Michel Gendreau, & Hossein Hashemi Doulabi
Programme: Program:	Mathématiques appliquées

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Sustainable Physical Internet with Uncertain Flow-Dependent Travel Times: Modeling and Solution Approach

MARJAN SADEGHI
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques appliquées

Décembre 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Sustainable Physical Internet with Uncertain Flow-Dependent Travel Times: Modeling and Solution Approach

présenté par **Marjan SADEGHI**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Antoine LEGRAIN, président
Michel GENDREAU, membre et directeur de recherche
Hossein HASHEMI DOULABI, membre et codirecteur de recherche
Walter REI, membre

DEDICATION

To my parents, thank you for your boundless love and sacrifices that shaped my world. To my brother, not just a sibling but my best friend, whose companionship and laughter light up the days.

ACKNOWLEDGEMENTS

I would like to extend my heartfelt thanks to Drs. Michel Gendreau and Hossein Hashemi Doulabi, for enabling me to undertake this research. I greatly appreciate their dedication to my academic and professional growth. Thank you both for your unwavering support and belief in this work.

RÉSUMÉ

L'Internet physique (PI) représente un cadre logistique à la pointe de la technologie conçu pour améliorer l'efficacité et le développement durable des chaînes d'approvisionnement mondiales grâce à l'utilisation de conteneurs modulaires standardisés et de systèmes interconnectés. Ce mémoire examine comment les principes de développement durable, économiques, environnementaux et sociaux peuvent être intégrés efficacement dans les réseaux PI, en se concentrant particulièrement sur les effets des temps de déplacement incertains et dépendant du flux, ainsi que sur la complexité des opérations de chargement et de déchargement. L'objectif est d'optimiser l'utilisation des ressources, de réduire les coûts et d'améliorer la responsabilité environnementale grâce à la conteneurisation modulaire au sein des réseaux PI.

Nous introduisons un nouveau modèle de programmation en nombres entiers mixte pour traiter ces questions, intégrant la capacité des liaisons comme une variable stochastique affectée par des facteurs tels que les catastrophes naturelles et l'entretien. Notre modèle aborde également les incertitudes de temps de déplacement en utilisant la fonction du Bureau of Public Roads (BPR), qui relie les temps de déplacement à la capacité et au flux de trafic. Cette approche globale permet une analyse approfondie des opérations PI, y compris une gestion détaillée des processus de chargement et de déchargement.

Nous utilisons le solveur BARON pour résoudre des problèmes de taille (jusqu'à 10 nœuds), mais les problèmes plus importants présentent des défis computationnels significatifs. Pour aborder ces problèmes plus importants, nous mettons en œuvre deux algorithmes heuristiques : les Algorithmes Génétiques (GA) et les Algorithmes Compétitifs Impérialistes (ICA). Les résultats indiquent que, bien que les méthodes exactes réussissent à résoudre les instances plus petites, le GA surpasse constamment l'ICA en termes de qualité de solution et d'efficacité computationnelle pour les problèmes plus grands. En particulier, le GA fournit des solutions supérieures à divers niveaux de perturbation, avec des temps de calcul raisonnables, tandis que l'ICA présente des limites dans les applications à grande échelle.

Les résultats mettent en évidence l'impact essentiel de la variabilité des temps de déplacement et de la complexité des hubs sur l'optimisation des réseaux PI. Bien que la recherche apporte des éclairages précieux, des limitations telles que les simplifications de modélisation et les défis liés à la disponibilité des données sont reconnues. Les futures directions de recherche incluent l'incorporation de prévisions de demande avancées, l'amélioration de l'évolutivité du modèle et la validation du modèle proposé dans des scénarios réels. De plus, l'intégration potentielle de technologies émergentes telles que la blockchain et l'Internet des objets (IoT)

est discutée pour améliorer l'adaptabilité dans des environnements logistiques dynamiques.

ABSTRACT

The Physical Internet (PI) represents a cutting-edge logistics framework designed to improve efficiency and sustainability in global supply chains through standardized modular containers and interconnected systems. This thesis investigates how economic, environmental, and social sustainability principles can be effectively integrated into PI networks, focusing on the effects of uncertain, flow-dependent travel times on the intricacies of loading and unloading operations. The aim is to optimize resource utilization, reduce costs, and improve environmental responsibility through modular containerization within PI networks.

We introduce a novel mixed-integer programming model to tackle these issues, incorporating link capacity as a stochastic variable affected by natural disasters and maintenance factors. Our model also addresses travel time uncertainties by utilizing the Bureau of Public Roads (BPR) function, which links travel times to capacity and traffic flow. This comprehensive approach allows for an in-depth analysis of PI operations, including detailed handling of loading and unloading processes.

We use the BARON solver to address smaller-scale problems (up to 10 nodes), but larger instances present significant computational challenges. We implement two heuristic algorithms to tackle these larger problems: Genetic Algorithms (GA) and Imperialist Competitive Algorithms (ICA). Specifically, the GA delivers superior solutions across varying levels of disruption, achieving reasonable computational times, while the ICA demonstrates limitations in larger-scale applications.

The results highlight the significant impact of travel time variability and hub complexity on the optimization of PI networks. Although the research contributes valuable insights, limitations such as modeling simplifications and data availability challenges are acknowledged. Future research directions include incorporating advanced demand forecasting, enhancing model scalability, and validating the proposed model in real-world scenarios. Additionally, the potential integration of emerging technologies like blockchain and the Internet of Things (IoT) is discussed to improve adaptability in dynamic logistics environments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

PI	Physical Internet
SCM	Supply Chain Management
GA	Genetic Algorithm
BPR	Bureau of Public Roads
MILP	Mixed-Integer Linear Programming
HGA-VNS	Hybrid Genetic Algorithm - Variable Neighborhood Search
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
MAPE	Mean Absolute Percentage Error
RMSE	Root Mean Square Error
ARIMAX	AutoRegressive Integrated Moving Average with Explanatory Variables
SVR	Support Vector Regression
MLR	Multiple Linear Regression
VMI	Vendor-Managed Inventory
MAS	Multi-Agent System
RL	Reinforcement Learning
DT	Digital Twin
IoT	Internet of Things
MOW	Meals-on-Wheels
FCN	Fast Consumer Networks

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Urbanization, rising populations, and increasing demands in cities make addressing city logistics essential for managing the complex challenges they pose. City logistics involves the efficient management of goods, services, and information within urban areas to balance economic growth, environmental sustainability, and quality of life. The final leg of the supply chain, often referred to as last-mile delivery, is a critical component of city logistics. Within the framework of city logistics, this phase focuses on optimizing the movement of goods from urban distribution centers, warehouses, or consolidation hubs to their final destination. In Canada, freight transportation including last-mile delivery contributed 3.9% of GDP in 2022, amounting to $81.4 billion in economic activity[1].

Last-mile delivery in city logistics is essential for modern economies, facilitating the efficient and timely movement of goods. It drives economic productivity, fosters trade, and boosts local businesses and e-commerce growth. By addressing challenges such as congestion and pollution, last-mile delivery ensures sustainable urban development, optimizes resource allocation, and fosters thriving and resilient communities. As a critical component of city logistics, last-mile delivery bridges the gap between regional distribution centers and end-users, ensuring that goods reach their final destinations efficiently and sustainably.

Benoit Montreuil proposed the concept of the Physical Internet (PI), advocating for modular, standardized, and interoperable freight systems to improve efficiency and sustainability in logistics [1]. This thesis is particularly relevant to last-mile delivery in city logistics, where PI can address urban delivery challenges effectively. PI is a comprehensive supply chain management concept that encompasses key aspects, such as efficiency and sustainability [2]. PI is an open global logistics system based on physical goods, digital platforms, and operational processes, all of which are standardized by protocols for modular encapsulation, collaboration, and smart interfaces. In PI, the goals are achievable only by considering the concept of "modular containers" [3]. Hence, objects in PI are handled, moved, stored, supplied, and used more efficiently and effectively with standardized containers.

Efficiency within the PI emphasizes refining processes and operations to reduce waste and enhance resource utilization. This includes minimizing transportation costs by using modular containers and reducing empty trips. The PI is crucial in terms of saving money based on its potential to significantly reduce logistics costs, by emphasizing the efficient movement of goods using modular containers. By maximizing loads and minimizing empty vehicle runs, it promotes cost-effective transportation solutions for goods and services. Furthermore, PI

[1] `https://tc.canada.ca`

reduces delays and minimizes operational costs by optimizing routes using traffic flow and travel time considerations resulting in a decrease in empty runs. By strategically placing hub facilities, they might enable the consolidation of shipments, which could result in higher vehicle loads and maximize transportation efficiency. This consolidation not only lowers fuel consumption and emissions but also enhances asset utilization, reducing overall logistics expenses.

Efficiency in the PI is enhanced by integrating various entities within the supply chain, such as retailers, distribution hubs, and customers. This collaboration allows for streamlined processes, improved communication, and better resource utilization, ultimately leading to reduced costs and faster delivery times. By working together, supply chain participants can optimize logistics and create a more responsive supply chain. The PI emphasizes the flow of goods through standardized containers and efficient logistics networks, aiming to optimize resource use while reducing environmental impact. As a result, the implementation of the PI offers significant cost-saving potential, with estimates suggesting reductions of up to 30% on global logistics expenses, potentially translating to trillions of dollars [2, 4].

Sustainability in the PI encompasses three main pillars: social, economic, and environmental [1]. Social sustainability in the context of PI refers to the enhancement of societal well-being through more equitable and efficient logistics and supply chain operations. A crucial aspect of this is reducing the travel time for drivers, which significantly impacts social sustainability [1]. Designing a PI transportation network that prioritizes minimizing travel time makes the model more socially beneficial. By optimizing travel time in links and routes, PI minimizes long hours on the road, reducing fatigue and improving the overall quality of life for drivers [1]. Retailers receive goods more promptly, hubs operate more smoothly, and customers enjoy faster delivery times.

Economic sustainability, coupled with the feature of standardized modular containers, represents a transformative approach to logistics and transportation systems. Standardization in container sizes and modular units within supply chains underpins efficiency gains and cost reductions. By adopting standardized modular containers, businesses can optimize packing density, streamline handling processes, and enhance compatibility across different modes of transport, such as road, rail, sea, and air [1]. This uniformity not only simplifies logistics operations but also reduces the need for specialized handling equipment and facilities, leading to significant savings in operational costs. Reduced transportation costs and improved asset utilization contribute to lower overall logistics expenses, enhancing profitability while maintaining competitive pricing in global markets [1].

Environmental sustainability in the PI is significantly bolstered by modular containers. Modular containers support scalability and flexibility in logistics operations by adapting to diverse

cargo sizes and types, thereby reducing waste. By maximizing space efficiency and minimizing the need for additional packaging and empty vehicles, they reduce fuel consumption and greenhouse gas emissions associated with transport operations [1]. Efficient logistics through modular containerization means fewer vehicles on the road, which translates to reduced congestion and smoother traffic flow [1]. Furthermore, modular containers support a circular economy by enabling easier recycling and refurbishment, thus extending their usability and reducing overall waste generation [1]. This integration into PI exemplifies a commitment to sustainable supply chain practices, emphasizing resource efficiency and environmental responsibility. There are several complicated factors in the way of achieving the above efficiencies and sustainability. Two of the most important challenges are the complexity of loading and unloading processes, and also uncertain flow-dependent travel times.

The uncertainty in travel times within PI has significant implications for sustainability across economic, environmental, and social aspects. Economically, unpredictable travel durations can lead to increased operational costs due to higher fuel consumption and extended vehicle maintenance needs, which ultimately erodes profit margins and affects the competitiveness of businesses. Environmentally, delays in transportation contribute to higher greenhouse gas emissions as vehicles remain on the road longer than necessary, undermining efforts to minimize the ecological footprint of logistics operations. Socially, the impact of longer travel times extends to driver well-being, as extended hours on the road can lead to fatigue, reduced job satisfaction, and increased safety risks. By disrupting the timely flow of goods, these delays not only affect service reliability but also diminish the overall quality of life for drivers and customers alike. Therefore, effectively addressing the uncertainty of travel times is crucial for enhancing sustainability in the PI, ensuring that economic efficiency, environmental responsibility, and social well-being are all prioritized.

Travel times in PI networks are typically flow-dependent, meaning that they are influenced by the volume of traffic on the road. In the context of the PI, this uncertainty in travel time significantly impacts the efficiency, reliability, and overall effectiveness of logistics operations. Firstly, unexpectedly longer travel times directly impacts the timely transportation of goods between different points in the network. Secondly, operational efficiency suffers when unexpected delays disrupt supply chain operations, leading to more time spent idling or stuck in traffic for vehicles. This not only reduces the overall productivity of drivers and vehicles but also increases driver fatigue. Thirdly, operational costs rise due to extended travel duration, leading to higher fuel consumption and potential maintenance expenses. Additionally, there is an increased risk of goods being damaged during transit, which undermines product quality and customer satisfaction. Lastly, service reliability suffers as unpredictable delays jeopardize meeting delivery commitments, potentially harming relationships with clients.

Therefore, ignoring the flow-dependency of travel times results in various inefficiencies and unsustainability in PI networks.

Loading and unloading processes in the PI are pivotal concepts that focus on seamless transitions of goods between vehicles and their efficient handling [1]. Aligning loading and unloading operations with vehicle travel time is essential for an efficient and reliable PI network. Standardized modular containers enable easy handling and rapid transfers, reducing the need for reloading at hubs. When loading and unloading are effectively managed, vehicles can maintain more predictable travel times and move continuously through the network. This is essential for minimizing delays at hubs. Moreover, smooth transitions between different transport modes ensure that each vehicle spends minimal time idle or waiting. Therefore, considering loading and unloading operations with vehicle travel time process minimizes queuing, limits downtime, and improves the throughput of goods across the network. By coordinating vehicle travel time with loading and unloading, PI networks foster a well-paced flow of goods, meeting delivery windows with greater accuracy, enhancing service reliability, and improving overall efficiency within the logistics ecosystem.

Efficient loading and unloading in the PI are vital for achieving sustainability across environmental, economic, and social dimensions. In terms of environmental benefits, streamlined handling processes reduce vehicle idle time, which directly decreases fuel consumption and greenhouse gas emissions. From an economic perspective, optimized loading and unloading lower operational costs in several ways. Fast and predictable handling reduces labor hours needed for transfers and lowers fuel costs due to minimized idle time. Socially, efficient loading and unloading processes improve vehicle waiting times, reduce driver fatigue, and improve job satisfaction by creating more predictable schedules, minimizing time away from home, and enhancing safety and well-being.

In the literature, several works address various aspects of sustainability and travel time uncertainty in links. In the context of sustainability, Luo et al. [5] focus on both operational efficiency and environmental sustainability through innovative PI logistics practices. Peng et al. [6] address sustainability in resilience planning to enhance environmental management through resource optimization and ensure economic viability with cost-effective strategies. They also promote social sustainability by safeguarding community interests in supply chain resilience efforts. Li et al. [7] tackle sustainability in container transportation by optimizing routes for environmental impact reduction, accounting for social sustainability impacts on stakeholders and communities on stakeholders and communities. Zheng et al. [8] assess PI urban logistics, focusing on environmental impact reduction, economic efficiency, and social welfare improvements in urban areas. Leung et al. [9] emphasize sustainability by optimizing resources, ensuring economic viability through efficient strategies, and enhancing social

sustainability via improved logistics operations and community impacts.

Lin et al. [10] integrate sustainability considerations in PI-enabled meal delivery systems, aiming to enhance environmental stewardship, ensure economic viability through cost-effective operations, and promote social sustainability by improving service accessibility and community impacts. Also, in the literature review, from a travel time uncertainty perspective, Chargui et al. [11] focus on this topic in PI networks using probabilistic methods. They consider that disruptions such as traffic congestion can lead to delays in trucks' arrival times. Pan et al. [12] aim to enhance the efficiency and reliability of perishable product bundling by addressing the challenges posed by travel time uncertainty through advanced modeling, real-time data utilization, and optimization techniques in PI networks.

Essghaier et al. [13] consider leveraging fuzzy logic and triangular fuzzy numbers to effectively manage and mitigate the uncertainty associated with truck arrival times in cross-docking operations in the PI networks. Liu et al. [14] adopt a comprehensive approach to address uncertainties in logistic hub capacity planning for relay transportation in PI. It incorporates stochastic modeling and scenario analysis to account for variability in demand and travel times in the PI network. Liu et al. [15] emphasize the importance of integrating historical data, probabilistic modeling of demand, and scenario-based approaches for managing travel time uncertainties in transportation hub capacity planning in PI networks.

The above studies collectively highlight the multidimensional benefits of integrating PI principles to advance sustainability in logistics and supply chain operations. However, none of these works have focused on all three aspects of sustainability including economic, environmental, and social while considering flow-dependent travel times and loading unloading operational details in PI networks. In this research, we study a PI problem with different types of sustainability while addressing flow-dependent travel times and loading/unloading operational details.

In this thesis, we wish to examine how PI principles can be applied in a specific setting. Namely, we consider the case of a city logistic operator controlling flows of vehicles transporting goods over the metropolitan area. This operator tries to optimize flows and operations to meet the needs and goals of several stakeholders: retailer companies, shippers, city government, representatives of popular groups, etc. We propose a high-level optimization model to determine the flows of goods between origins and destinations, with intermediate hub consolidation. In this model, nodes represent the origins and destinations of freight flows, while transshipment edges correspond to virtual lines between the nodes (e.g. a link may represent a sub-network corresponding to the shortest path connecting different locations).

We propose a novel Mixed-Integer Programming (MIP) model to formulate this problem. In our model, we introduce link capacity as a random variable influenced by various factors such

as natural disasters and maintenance activities. Additionally, we consider flow-dependent travel time by linking them to capacity and traffic flow using the BPR function. Travel time thus becomes another random variable in our mathematical model that depends on the link capacity and flows in the link. Moreover, our model is more advanced than other models in the literature as it formulates the details of loading/unloading operations in the PI network. Since the proposed model is capable of handling small instances of the sustainable PI problem, we develop a Genetic Algorithm (GA) to address larger instances.
The main contributions of our work are as follows:

- In the context of PI, we study a problem with all dimensions of sustainability, including economic, social, and environmental.
- We introduce uncertainty in link capacity, flow dependency of travel times, and the details of loading and unloading processes. These features represent novel contributions to the existing PI literature.
- We have developed a novel mixed-integer programming mathematical model with chance constraints for the above PI problem. We also proposed a GA as the solution method for large instances.
- We present computational results for the proposed mathematical model and the GA.

The remainder of this thesis is organized into five chapters, each addressing a specific aspect of the research problem and its solution. Chapter 2 provides a comprehensive review of the relevant literature. Chapter 3 elaborates on the problem statement, key assumptions, and the proposed mathematical model. Chapter 4 describes the solution methodology in detail. Chapter 5 presents the results and computational analysis. Finally, Chapter 6 concludes the research by summarizing the key findings, discussing limitations, and proposing potential directions for future research.

CHAPTER 2 LITERATURE REVIEW

In this section, we review the literature on the PI, focusing on optimization. The PI is a relatively new but rapidly expanding field with strong potential for logistics and supply chain management. We organize existing studies in PI by key applications, such as routing, inventory planning, and location-allocation, and highlight how each paper addresses these topics within the PI framework. Our literature review emphasizes the main aspects of each paper, especially how they handle travel time uncertainty, capacity uncertainty, loading and unloading, and sustainability—core contributions of our thesis. We also review the foundational assumptions in the reviewed studies, along with the mathematical methods used to address uncertainties. The final section of this literature review is organized into two subsections: Uncertainty Handling and Optimization Strategies, and Heuristic and Meta-heuristic Approaches. This comprehensive analysis provides valuable insights into the advancements in PI optimization and highlights its potential to transform logistics.

2.1 Routing and Location-Allocation

Venkatadri et al. [16] model the impact of shipment consolidation in PI logistics on transportation and inventory costs, comparing it with traditional logistics systems. They focus on optimizing transportation routes and consolidating shipments. Routing involves making decisions on how shipments are consolidated and moved through the network to minimize transportation costs and maximize efficiency. Uncertainty is not explicitly modeled, and the paper assumes deterministic demand and supply flows. Key assumptions include the modularization of shipments and a hub-based PI network for consolidation. The paper develops a Mixed-Integer Linear Programming (MILP) model to optimize point-to-point dispatches.
Puskas et al. [17] address a logistics routing problem, focusing on optimizing platooning and vehicle exchanges between platoons at virtual hubs. Platooning means vehicles travel closely to reduce aerodynamic drag and improve fuel use. They develop models for re-configuring two platoons, which involves the strategic exchange of vehicles between platoons at a virtual hub. This reconfiguration aims at optimizing transportation efficiency and reducing costs.
Regarding travel time uncertainty, these two works adopted two different approaches. Venkatadri et al. [16] address travel time uncertainty through adaptive routing strategies. They employ dynamic optimization to adjust routes in real-time. On the other hand, Puskas et al. [17] use reinforcement learning for more flexible adaptation to changing conditions to handle travel time uncertainty through real-time feedback, making it particularly effective for large, com-

plex networks. While both methods enhance travel time reliability, reinforcement learning provides a more scalable solution, whereas dynamic optimization may be a better fit for smaller, cost-sensitive setups.

Fazili et al. [18] compare PI logistics with traditional and hybrid models, focusing on routing efficiency in Eastern Canada's road network. In the hybrid system, consolidation occurs at the source and destination PI hubs. Containers are not repeatedly packed or unpacked at each PI hub along the route. This model offers partial consolidation benefits from PI without the extensive reloading processes. They aim to balance the efficiency of PI with the simplicity of conventional logistics. They analyze PI usage of modular containers and shorter-haul routes versus traditional long-haul methods to assess the benefits and drawbacks of each. By optimizing truck and driver routes, the study highlights PI's potential to improve environmental sustainability and resource efficiency. Their findings suggest that while traditional models meet some needs, PI could be more sustainable and efficient, especially for operations focused on environmental impact and flexible routing.

In addressing road capacity uncertainty, both Fazili et al. [18] and Venkatadri et al. [16] explore how fluctuating traffic and road conditions impact logistics performance in PI and hybrid systems, though each focuses on different aspects. Fazili et al. [18], in their comparison of conventional, hybrid, and PI-based logistics, highlight that variability in road capacity can disrupt delivery times. This variability can also increase fuel consumption and decrease overall efficiency, especially in PI systems where multiple short trips between hubs are common. On the other hand, Venkatadri et al. [16] examine the effects of road capacity on PI logistics, focusing specifically on load consolidation and transportation costs. Both studies show that PI systems offer environmental and operational benefits but are very sensitive to changes in road capacity, which can affect how well they work.

Pal et al. [19] address the challenges associated with fresh food transportation, focusing on maximizing product freshness and minimizing waste. The problem described in the paper encompasses elements of both routing and location allocation. Shared infrastructure and modular operations form the basis of the PI network. They identify issues such as inefficient logistics, high carbon footprint, and extended driver away-from-home times as significant problems in the current distribution networks. The model incorporates freshness as a decaying variable over time, indirectly accounting for delivery time uncertainty. Besides, Luo et al. [5] research can be categorized as a logistics optimization problem, specifically focusing on routing and resource allocation within the context of customized furniture delivery in metropolitan areas using the PI. A modified vehicle routing problem model with simultaneous pickups and deliveries is applied, supported by sensitivity analysis to test the robustness of the model. They develop a new mathematical modeling to maximize profits for shippers

while ensuring stable incomes for carriers. They validate their approach through a real-life case study. The paper discusses the impact of demand volatility and rising fuel costs as uncertain parameters in the delivery model.

In terms of loading and unloading Pal et al. [19] highlight how modular containers streamline fresh food transportation, improving handling efficiency, reducing labor, and optimizing space, which is crucial for maintaining product freshness. Luo et al. [5] demonstrate how smart containers enhance operational efficiency by reducing the time spent on loading and unloading. However, they note that irregularly shaped items, such as furniture, still pose loading challenges despite the benefits of modular systems. Both studies also point out that implementing these advanced systems requires a robust IT infrastructure, which can raise operational costs and complexity. Pal et al. [19] examine how transportation efficiency impacts product freshness. Faster and more efficient handling does not always ensure that freshness is preserved optimally. On the other hand, Luo et al. [5] highlight efficiency gains but note the logistical challenge of managing bulky, varied items within modular limits.

Chadha et al. [20] focus on optimizing routes in the Mexican automotive supply chain by combining traditional peddling methods with PI frameworks. They developed three models. Model P is a fully PI-based system utilizing smart technologies and modular resources. Model S uses traditional peddling to consolidate deliveries, maximizing vehicle use but extending travel distance. Model H is a hybrid approach that combines the collaborative efficiencies of PI with the capacity gains of peddling. Peddling is a concept in which multiple deliveries or pickups are combined into a single, consolidated route. Their study highlights routing optimization, sustainability, and resource efficiency but assumes deterministic demand without accounting for uncertainty. Moreover, Lin et al. [10] examine meal delivery logistics for the meals-on-wheels service, analyzing cost, and customer satisfaction trade-offs across self-delivery, outsourcing, and volunteer options. Their model integrates PI containers and smart lockers to optimize facility location and vehicle routing within time windows to ensure meal freshness. Unlike Chadha et al. [20], Lin et al. [10] address uncertainties in demand distribution and parking site availability.

In terms of sustainability, Chadha et al. [20], Fazili et al. [18], and Lin et al. [10] focus on sustainability improvements in logistics through consolidation strategies and dynamic route optimization. By leveraging PI models, these studies demonstrate that reducing empty miles and optimizing load capacity can significantly cut greenhouse gas emissions and fuel use. However, Chadha et al. [20] and Fazili et al. [18] both point out that the operational complexity of managing multi-stop consolidations in PI systems can lead to delays and inefficiencies. This complexity can potentially diminish some of the environmental benefits that PI aims to achieve.

In addition, Venkatadri et al. [16] emphasize that while PI systems enhance load consolidation, optimizing consolidation alone is insufficient to reduce greenhouse gas emissions effectively; additional strategies, such as efficient routing and scheduling, are essential for meaningful emission reductions across the logistics chain. Alongside this, Luo et al. [5] add another layer of complexity, noting that PI systems are highly sensitive to fuel price fluctuations, which could undermine the sustainability gains achieved through reduced emissions if external market conditions shift. Together, these studies reveal that while PI and consolidation strategies offer clear sustainability advantages, they also introduce vulnerabilities that must be managed carefully for optimal outcomes.

2.2 Inventory and Production Planning

Yang et al. [21] tackle inventory management issues in vendor-managed systems by using a PI framework for better demand responsiveness. They suggest a decentralized network where vendors manage inventory across locations, adapting stock to demand changes. This PI approach allows vendors to source widely and meet retailer needs reliably, lowering logistics costs. Yang et al. [21] support a more resilient, efficient inventory system to handle supply and demand uncertainties.

In terms of loading and unloading, Yang et al. [21] explore how flexible, decentralized PI networks can reduce loading and unloading delays. Their model improves flow and lowers logistics costs. However, it depends on advanced, real-time infrastructure and data integration. This requirement may restrict its use in areas with limited technology.

Ji et al. [22] and Peng et al. [6] both explore improvements in production-inventory-distribution systems. Ji et al. [22] develop a stochastic decision-making model that integrates production, inventory, and distribution planning to minimize costs and improve service levels. Their model uses mixed-integer linear programming to consolidate loads, dynamically route vehicles based on hub loading capacities, and handle demand and disruption uncertainties across PI hubs and standardized containers. This approach enhances efficiency and cuts costs, showing PI's potential to reduce fixed and variable vehicle expenses and remain resilient under changing conditions. However, it relies heavily on real-time data and robust IT infrastructure, which may limit its practicality in regions with limited digital resources. Moreover, Peng et al. [6] address a PI problem focusing on building resilience in production-inventory-distribution systems, against supply chain disruptions. The presented model includes both pre- and post-disruption strategies, allowing dynamic reconfiguration of logistics nodes.

In terms of travel time uncertainty Peng et al. [6] proposed a model that anticipates changes in travel time. It integrates strategies for before and after disruptions to keep production,

inventory, and distribution running smoothly. This approach builds resilience by preparing for delays; however, preplanned strategies may be less effective for sudden, unexpected disruptions. On the other hand, Yang et al. [21] address travel time uncertainty in their vendor-managed inventory model by a simulation-based optimization to adjust inventory levels across the PI network as conditions change. This improves service levels and reduces costs but relies on real-time data, which may be hard to access in regions with limited digital infrastructure. Both models handle travel time variability differently. Peng et al. [6] focuses on structured planning for disruptions, while Yang et al. [21] is more flexible for daily changes. However, both models require advanced data systems, which can be challenging to scale in less digital areas.

In terms of road capacity, Peng et al. [6] address road capacity uncertainty with a resilience model that includes strategies for both before and after disruptions. This approach supports long-term planning for capacity issues. However, it depends on preplanned actions, making it less adaptable to sudden, unexpected shortages, as it focuses mainly on anticipated scenarios. Peng et al. [23] address the optimization of transportation routing and planning within the context of an integrated production-inventory-distribution system enabled by the PI. This paper considers production, inventory, routing, and loading/unloading decisions. The study evaluates economic aspects by considering total costs, environmental impacts by assessing greenhouse gas emissions, and social factors by examining the implications of accident risks. In terms of sustainability, Peng et al. [23] address sustainability by improving vehicle utilization to cut emissions and transportation costs in a PI-enabled logistics model. By focusing on both environmental goals and operational efficiency, the model accounts for road capacity fluctuations to promote sustainable logistics. However, this multi-objective approach adds complexity, making it challenging for smaller operators with limited resources. Thus, while the model showcases PI's potential for greener logistics, its complexity may restrict broader use, highlighting a balance between sustainability and practical accessibility.

2.3 Miscellaneous

Guo et al. [24] propose an innovative platform for on-demand delivery that replaces fixed routes with a competitive, real-time auction system where carriers bid to fulfill delivery requests. Using Deep Q Networks, the system applies reinforcement learning to adapt dynamically to demand, traffic, and delivery uncertainties, creating a flexible and responsive delivery network. However, this model, while adaptive, introduces variability in delivery times. It also demands high computational power, which may be challenging for regions with limited technology.

In terms of travel time uncertainty, Guo et al. [24] address how the system dynamically adapts to changing traffic conditions and real-time demand fluctuations, optimizing delivery schedules. The advantage of this model is its ability to adapt to real-time variations, providing flexibility in handling uncertain travel times. However, this solution demands substantial computational power, making it challenging to implement in smaller or less technologically advanced regions.

Chargui et al. [25] present a multi-agent model for sustainable truck scheduling and container grouping in a Road-Rail PI hub. The study focuses on efficiently scheduling trucks and grouping containers onto rail wagons. Key goals include minimizing wagon use, reducing container travel distances within the hub, and ensuring trucks operate on time. This approach supports sustainability by cutting energy consumption, lowering costs, and reducing environmental impact.

Tan et al. [26] tackle parking space optimization in congested urban areas using a PI-enabled system. Their model dynamically allocates and prices public and private parking based on fluctuating demand, considering short-term and long-term drivers. A continuous-time Markov chain models arrival and departure uncertainties, while a reverse Vickrey auction sets prices. The goal is to maximize parking space use and stabilize pricing.

In terms of capacity uncertainty, Chargui et al. [25], Guo et al. [24], and Tan et al. [26] each address it through adaptive, data-driven models suited to specific logistics challenges. Chargui et al. [25] manage fluctuating capacity in a Road-Rail PI hub by dynamically rescheduling trucks and grouping containers in response to real-time disruptions, such as arrival delays. Guo et al. [24] focus on on-demand delivery, utilizing reinforcement learning to adjust delivery schedules in real-time based on demand and traffic variability. Tan et al. [26] handle urban parking capacity to dynamically allocate parking as availability shifts. Each model enhances flexibility under uncertain conditions but depends on substantial computational power, making these solutions most applicable to advanced, technology-enabled settings.

In terms of loading and unloading, both Chargui et al. [25] and Tan et al. [26] offer adaptive solutions, though in different settings. Chargui et al. [25] improve container grouping and truck scheduling in Road-Rail PI hubs. They consider disruptions in container availability and truck arrivals, which reduce logistics costs and internal travel distances. However, the system's high computational demands may limit its use for smaller logistics operators.

In terms of sustainability, both Chargui et al. [25] and Tan et al. [26] address environmental impact reduction in their respective fields using different approaches. Chargui et al. [25] improve sustainability by optimizing truck scheduling and container grouping, which lowers energy use and CO_2 emissions by reducing wagon numbers and container travel distances. Although effective, this method requires substantial computational resources, limiting its

broad application. Similarly, Tan et al. [26] enhance urban parking sustainability by reducing emissions by minimizing drivers' search times for parking. However, this approach also depends on advanced data systems, making it less feasible in regions with limited technology. Both models contribute to environmental sustainability. Chargui et al. [25] focus on logistics hubs, while Tan et al. [26] address urban parking. Nonetheless, both face scalability challenges due to their reliance on sophisticated technology.

2.4 Methodologies employed in optimizing the PI scope

2.4.1 Uncertainty Handling and Optimization Strategies

Puskás et al. [17] use reinforcement learning to handle dynamic conditions like changing vehicle numbers and dispatch intervals, allowing the system to adapt through continuous learning. While reinforcement learning adjusts well to new conditions, it requires large datasets and significant computational power. Besides, Guo et al. [24] deploy a multi-agent system in an open trading environment to handle uncertainties in demand and traffic through Deep Q Network-based reinforcement learning. The multi-agent system approach is highly effective in managing complex agent interactions but may encounter scalability issues as agent and transaction volumes increase, much like the computational challenges faced by Puskás et al. [17].

Peng et al. [6] adopt a two-stage stochastic programming model to anticipate and mitigate disruptions in production-inventory-distribution systems. This model provides robustness by integrating both Before-and-after event mitigation strategies, offering an anticipatory approach. However, the two-stage stochastic programming's reliance on advanced event data may limit its responsiveness to sudden disruptions, a disadvantage compared to the real-time adaptability of Reinforcement learning-based and multi-agent system approaches.

Yang et al. [21] propose a simulation-based optimization for vendor-managed inventory in PI systems, which manages stochastic demand with flexibility in real-time demand fluctuations. However, the high computational demands of simulations may limit scalability in large networks. Meanwhile, Fazili et al. [18] address road capacity uncertainty with Monte Carlo simulations within a three-phase optimization framework. While Monte Carlo methods offer an efficient way to evaluate uncertain variables, their effectiveness is limited by the high computational cost of repeated simulations.

Tan et al. [26], allocate parking spaces. This approach is computationally efficient, but Markov chains may miss rare, high-impact events, limiting effectiveness in volatile environments. Ji et al. [22] introduce a multi-objective MILP model. While comprehensive, this approach's computational intensity could hinder timely decision-making in highly dynamic

environments.

These methods share a common goal of managing uncertainties in PI-enabled systems through dynamic adaptability. Reinforcement learning and multi-agent systems offer high flexibility and real-time decision-making but are limited by heavy computational demands, making them less feasible for smaller operators. In contrast, Monte Carlo simulations and Markov chains are more computationally efficient but lack the same level of real-time adaptability.

2.4.2 Heuristic and Meta-heuristic Algorithms

Peng et al. [6] use a three-phase heuristic approach combined with a fix-and-optimize strategy. to enhance resilience in PI networks. While flexible for large-scale issues, their approach faces scalability challenges due to computational demands, making real-time application difficult in variable networks. Luo et al. [5] take a simpler approach, applying heuristics for customized furniture delivery. This method is efficient but may lead to suboptimal choices in dynamic PI systems requiring real-time flexibility. Also, Chadha et al. [20] implement integer programming with consolidation strategies to optimize freight routing within PI networks, using heuristics to enhance route precision and cost efficiency. Additionally, Grover et al. [27] studies on time-based parcel consolidation employ dynamic heuristic methods to optimize container loading in PI hubs, managing constraints related to departure schedules and container dimensions

Puskás et al. [17] combine reinforcement learning with heuristics for platoon reconfiguration, finding that heuristics perform well in low-traffic scenarios, while reinforcement learning is better suited for higher traffic conditions, achieving a balance between adaptability and computational efficiency. Yang et al. [21] employ simulated annealing for vendor-managed inventory (VMI) within PI, which supports global solution exploration but may struggle with computational loads in real-time data environments. Similarly, Ji et al. [22] integrate heuristics in a multi-objective mixed-integer model to optimize sustainability across economic, environmental, and social factors, but the complexity of multi-objective optimization limits its suitability for fast-moving, dynamic networks.

Chargui et al. [11] integrated a multi-agent system (MAS) with a hybrid meta-heuristic framework for truck scheduling and container grouping in Road-Rail PI hubs. By embedding concurrent meta-heuristics, their approach aimed to balance efficiency and adaptability, ensuring sustainable logistics through real-time responsiveness to disruptions. Pure heuristic methods, like those applied by Luo et al. [5], may result in less optimal performance in dynamic settings. Reinforcement learning, though computationally intense, provides excellent adaptability for real-time decision-making, as shown by Puskás et al. [17]. Hybrid approaches that combine mixed-integer linear programming with multi-agent systems or reinforcement

learning effectively blend rigor with flexibility, though they still encounter challenges with computational cost and scalability.
These studies illustrate diverse applications of heuristics and meta-heuristics in PI systems, emphasizing both computational efficiency and the need for scalability in real-time, adaptive environments.

2.5 Summary and Gap Analysis

The literature review on the PI systems reveals that substantial progress has been made in optimizing various aspects of logistics operations, such as routing, location-allocation, inventory, and production planning. Many studies demonstrate the benefits of PI-based systems, particularly in terms of sustainability, efficiency improvements, and operational flexibility. However, challenges remain, especially in handling uncertainties and the scalability of solutions. Key conclusions from the literature are presented below.

2.5.1 Loading and Unloading

Research on loading and unloading within PI hubs primarily focuses on efficiency improvements through modular, smart container systems. Studies by [19] and [5] explore the benefits of standardized containers, which reduce handling times and improve transportation efficiency for perishables and bulky items, respectively.
Most studies model loading and unloading under deterministic conditions, lacking consideration for real-world uncertainties such as travel time disruptions. According to the literature review, no one considers loading and unloading alongside flow-dependent uncertain travel time.

2.5.2 Sustainability

Sustainability is a prominent focus in PI logistics research, with systems demonstrating potential for reducing greenhouse gas emissions, optimizing routes, and minimizing fuel consumption. Studies such as [23] and [11] emphasize these environmental benefits, showcasing strategies like modular container use, reduced travel distances, and energy-efficient routing.
While sustainability is widely addressed, most models operate under deterministic assumptions without accounting for uncertainties such as capacity and travel time uncertainties. This thesis enhances sustainability modeling by integrating chance constraints and incorporating environmental unpredictability, ensuring operational reliability under fluctuating conditions. This approach extends beyond deterministic models by using the Bureau distri-

bution function to create a more flexible, environmentally responsible logistics solution. This thesis, by integrating travel time as an uncertain factor in both social and environmental objectives, contributes to the PI field.

2.5.3 Travel Time Uncertainty

Research generally addresses travel time uncertainty through adaptive routing strategies. Studies like [16] and [17] incorporate real-time adjustments based on changing traffic patterns, often leveraging reinforcement learning or dynamic optimization. Although effective, these models are computationally intensive, affecting scalability.

Existing studies rarely integrate flow-dependent travel time uncertainty, where traffic flow variations directly impact travel times. This thesis fills this gap by utilizing the BPR function and chance constraints, offering a comprehensive approach to travel time uncertainty by considering variable traffic flow. This ensures route reliability and adaptability, especially for larger PI networks facing unpredictable travel conditions.

2.5.4 Capacity Uncertainty

Capacity uncertainty is widely recognized as a significant factor impacting logistics networks, particularly within PI systems. Studies by [18] and [6] incorporate capacity variations using Monte Carlo simulations and stochastic programming, respectively, focusing on long-term resilience against predictable disruptions.

Capacity is frequently modeled as a fixed or known variable, limiting responsiveness to sudden, real-time capacity fluctuations. This thesis introduces capacity uncertainty by modeling it as a stochastic variable. By treating link capacity as a variable affected by environmental and operational conditions, this thesis enhances network reliability in response to unexpected capacity disruptions.

In conclusion, this thesis addresses several critical gaps identified in the PI logistics literature, specifically in the handling of loading and unloading, sustainability, uncertainties related to travel time, and capacity. By integrating stochastic elements across these domains, this work contributes a reliable adaptable framework that aligns with real-world logistics needs, advancing the optimization potential of PI systems in dynamic environments. This thesis's proposed approach aims to bridge these gaps by applying chance constraints and the Bureau distribution function, offering a balance between computational efficiency and reliability. By effectively addressing capacity uncertainties and travel time variability, this approach presents a scalable and adaptable solution for managing uncertainties in PI-enabled logistics systems.

CHAPTER 3 PROBLEM DESCRIPTION AND PROPOSED MATHEMATICAL MODELING

In this study, we address a logistics optimization problem within the PI framework, focusing on a three-layer network consisting of retailers, hubs, and customers. The objective is to design a distribution network in which products are efficiently routed through interconnected hubs to meet customer demands while addressing sustainability goals. The network is subject to capacity constraints and uncertainties affecting link capacity and travel time. These elements introduce challenges that require reliable decision-making in areas such as hub location-allocation and loading-unloading operations. We aim to optimize these aspects to achieve economic efficiency, environmental responsibility, and social benefits, balancing costs with carbon emission reductions and transportation time improvements. By integrating uncertain, flow-dependent travel times and capacity constraints, this research seeks to develop a model that supports sustainable logistics operations within an interconnected PI network.

3.1 Problem Statement

In this study, we investigate a PI problem with three layers including retailers denoted by $i \in I$, hubs denoted by $k, m \in H$, and customers denoted by $j \in J$. The reason for using two indices for hubs, $k, m \in H$, instead of just one is to clearly distinguish between the different hubs involved in a route when a vehicle travels through two hubs. Let V denote the set of all homogeneous vehicles, indexed by v. Each customer j has a determined demand from each retailer i, denoted by W_{ij}. We consider a finite number of vehicles (Nv) with a limited capacity (Cap) for transporting products in the distribution network. Hubs in this PI distribution network collect, switch, and consolidate demands. Each hub has a limited capacity (Ch_k) as well. Furthermore, each link in the network has a limited capacity (i.e., $\tilde{C}_{ik}$, $\tilde{C}_{km}$, $\tilde{C}_{mj}$). Notably, the capacity of each link can vary due to disruptions, making them inherently random variables denoted by a tilde sign. Moreover, The durations of travel times on links, denoted by $\tilde{T}_{ik}$, $\tilde{T}_{km}$, $\tilde{T}_{mj}$ are influenced by traffic volume on links and also the random capacity of links.

Here, we will discuss the decisions in the presented mathematical model. For simplicity, we divide them into two categories: location-allocation decisions and loading-unloading decisions. The first category includes decisions related to the location of hubs denoted by binary variables y_k the allocation of retailers and customers to hubs denoted by binary variables z_{ik}, z_{mj}, and the amount of originated flows in each link denoted by continuous variables x_{ik},

x_{km}, x_{mj}. Another decision within this category is the determination of routes between each retailer i and each customer j connected with at most two hubs k and m, which is indicated by binary variables x_{ijkm}. Also, in our problem, the expected value of travel time of links that are denoted by $e(\tilde{T}_{ik})$, $e(\tilde{T}_{km})$, and $e(\tilde{T}_{mj})$ are auxiliary decision variables, since travel times on links depend on the amount of flow over them.
The second category encompasses the loading-unloading decisions. The primary decision involves determining the amount of loading in each vehicle for each link. We denote the loading decision variables by l^v_{ik}, l'^v_{ikm}, and l''^v_{imj} as defined precisely in the Table of decision variables under section 3.1 on the links.
In this problem, the objective function addresses three types of sustainability economic, environmental, and social. The economic component of the objective function includes the hub establishment cost (F_k), and the transportation cost. The environmental component aims at minimizing carbon emission function which is based on the amount of loading in vehicles and travel time of vehicles which is used in equation (3.2). The social component minimizes the total transportation time.

3.2 Assumptions

The main assumptions in our problem are as follows:

- Travel times on roads depend on the amount of flow on links and are governed by the BPR functions.
- The network is fully connected between hubs.
- Vehicles are homogeneous.
- Demands (W_{ij}) are deterministic.
- Each route involves at least one hub and at most two hubs.
- Demands are measured in terms of the number of standard containers.

3.3 Notations

In our notation, the tilde sign indicates a random variable. We represent decision variables with lower-case letters and parameters with upper-case letters. For expectation, we maintain the same type of notation. This means that $e(\tilde{T}_{ik})$, $e(\tilde{T}_{km})$, $e(\tilde{T}_{mj})$, $e(\tilde{T}_{ijkm})$ are decision variables and are shown with lower-case letters. However, we denote $E(\frac{1}{\tilde{C}_{ik}^{Nb}})$ as expectations with upper-case letters, as it is a constant. For ease of reading the next section, we have summarized the notations as follows.

Sets and Indices:	
I	Set of retailers
J	Set of Customers
H	Set of Hubs
N	Set of all nodes $N = I \cup H \cup J$
V	Set of all vehicles in homogeneous fleet
i	Indices for retailers
j	Indices for customers
k, m	Indices for hubs
v	Indices for vehicles

Parameters:	
Nv	Number of vehicles.
P	Number of hubs.
∂	The maximum acceptable difference between the number of retailers and Customers allocated to opened hubs.
F_k	Fixed setup cost for hub k.
Fv_{ik}	Transportation cost in the link (i,k).
Fv_{km}	Transportation cost in the link (k,m).
Fv_{mj}	Transportation cost in the link (m,j).
Ch_k	Capacity of hub k.
α_{ik}	Link capacity exceeding probability for the link (i,k).
α_{km}	Link capacity exceeding probability for the link (k,m).
α_{mj}	Link capacity exceeding probability for the link (m,j).
θ_{ik}	Disruption multiplier of link (i,k).
θ_{km}	Disruption multiplier of link (k,m).
θ_{mj}	Disruption multiplier of link (m,j).
$\bar{T}_{ik}$	The travel time on the link (i,k) without considering the disruption.
$\bar{T}_{km}$	The travel time on the link (k,m) without considering the disruption.
$\bar{T}_{mj}$	The travel time on the link (m,j) without considering the disruption.
$\bar{C}_{ik}$	The nominal capacity without considering disruption in the link (i,k).
$\bar{C}_{km}$	The nominal capacity without considering disruption in the link (k,m).
$\bar{C}_{mj}$	The nominal capacity without considering disruption link (m,j).
$\tilde{C}_{ik}$	Capacity for the link (i,k) considering the effect of disruption.
$\tilde{C}_{km}$	Capacity for the link (k,m) considering the effect of disruption.
$\tilde{C}_{mj}$	Capacity for the link (m,j) considering the effect of disruption.
W_{ij}	The amount of demand by customer i from retailer j.
Cap	Capacity of each vehicle.
M	A very large and fixed number.
Nb	BPR road factor parameter.
β	Calibration parameter of BPR function.

Decision Variables:	
$e(\tilde{T}_{ik})$	The expected travel time on the link (i,k) considering the effect of disruption.
$e(\tilde{T}_{km})$	The expected travel time on the link (k,m) considering the effect of disruption.
$e(\tilde{T}_{mj})$	The expected travel time on the link (m,j) considering the effect of disruption.
$e(\tilde{T}_{ijkm})$	The expected travel time on route (i,j,k,m) considering the effect of disruption.
x_{ik}	Flow (in terms of the number of vehicles) between retailer i and hub k.
x_{km}	The flow (in terms of the number of vehicles) between hub k and hub m.
x_{mj}	The flow (in terms of the number of vehicles) between hub m and customer j.
x_{ijkm}	1 if there is any flow from retailer i to customer j through hubs k and m; 0 otherwise.
z_{ik}	1 if retailer i is allocated to hub k, 0 otherwise.
y_k	1 if hub k is established as a hub.
z_{mj}	1 if customer j is allocated to hub m, 0 otherwise.
b	The maximum number of customers allocated to any opened hub.
b'	The minimum number of customers allocated to any opened hub.
l_{ik}^{v}	The number of products in vehicle v traveling in the link (i,k).
$l_{ikm}^{\prime v}$	The number of products originated from retailer i traveling in link (k,m) with vehicle v.
$l_{imj}^{\prime\prime v}$	The number of products originated from retailer i traveling in the link (m,j) with vehicle v.

3.4 Components of Objective Function

We formulate the economic part of the objective function as follows:

3.4.1 Economic Objective Function

Here an economic part of the objective function is presented.

$$z_{eco} = \sum_{k\in H} F_k y_k + \sum_{i\in I}\sum_{k\in H} Fv_{ik}x_{ik} + \sum_{k\in H}\sum_{m\in H} Fv_{km}x_{km} + \sum_{m\in H}\sum_{j\in J} Fv_{mj}x_{mj} \quad (3.1)$$

The first term of the objective function represents the cost of establishing hubs. The remainder of this objective function presents the transportation costs.

3.4.2 Environmental Objective Function

Here, we present the environmental component of the objective function.

$$\begin{aligned} z_{env} &= \zeta_{env}[\sum_{i\in I}\sum_{k\in H}\sum_{v\in V}(\psi\cdot e(\tilde{T}_{ik}) + \eta\cdot l_{ik}^{v}) + \sum_{i\in I}\sum_{k\in H}\sum_{m\in H}\sum_{v\in V}(\psi\cdot e(\tilde{T}_{km}) + \eta\cdot l_{ikm}^{'v}) \\ &+ \sum_{i\in I}\sum_{m\in H}\sum_{j\in J}\sum_{v\in V}(\psi\cdot e(\tilde{T}_{mj}) + \eta\cdot l_{imj}^{''v})] \end{aligned} \quad (3.2)$$

In equation 3.2, we use ζ_{env} as a conversion parameter from kilogram to cost, as we aim to combine the three components of the objective function into a single function. The environmental objective function calculates the total greenhouse gas (GHG) emissions based on the amount of loading in each vehicle and the travel time of vehicles, using a carbon emission function ω. In this function, we considered that the carbon emission rate increases linearly with time and loading. Therefore, we defined the carbon emission function as $\omega(e(\tilde{T}_{ik}), l_{ik}^{v}) = \psi\cdot e(\tilde{T}_{ik}) + \eta\cdot l_{ik}^{v}$.The ψ and η are two constants that represent the emission rate per minute of travel time (kg/min), and the emission rate per container loaded on the vehicle (kg/container).

3.4.3 Social Objective Function

We present the social objective function as follows:

$$\begin{aligned} z_{soc} &= \zeta_{soc}[\sum_{i\in I}\sum_{k\in H} e(\tilde{T}_{ik}) + \sum_{k\in H}\sum_{m\in H} e(\tilde{T}_{km}) \\ &+ \sum_{m\in H}\sum_{j\in J} e(\tilde{T}_{mj})] \end{aligned} \quad (3.3)$$

The social objective function is determined by the total travel time in all network links. We use ζ_{soc} as a conversion parameter from time to cost. We note that the objective function is nonlinear due to the multiplication of decision variables.

3.5 Proposed Mathematical Model

In the following, we present the proposed mathematical model.

$$Minz = z_{eco} + z_{env} + z_{soc} \tag{3.4}$$

$Constraints$ $(3.1) - (3.3)$.

$$\sum_{k \in H} z_{ik} = 1 \quad \forall i \in I \tag{3.5}$$

$$\sum_{m \in H} z_{mj} = 1 \quad \forall j \in J \tag{3.6}$$

$$z_{ik} \leq y_k \quad \forall i \in I, k \in H \tag{3.7}$$

$$z_{mj} \leq y_m \quad \forall m \in H, j \in J \tag{3.8}$$

$$\sum_{k \in H} y_k = P \tag{3.9}$$

$$b \geq \sum_{i \in I} z_{ik} - M \cdot (1 - y_k) \quad \forall k \in H \tag{3.10}$$

$$b' \leq \sum_{i \in I} z_{ik} + M \cdot (1 - y_k) \quad \forall k \in H \tag{3.11}$$

$$b - b' \leq \partial \tag{3.12}$$

$$z_{ik} = \sum_{m \in H} x_{ijkm} \quad \forall i \in I, k \in H, j \in J \tag{3.13}$$

$$z_{mj} = \sum_{k \in H} x_{ijkm} \quad \forall i \in I, m \in H, j \in J \tag{3.14}$$

$$\sum_{k \in H} \sum_{m \in H} x_{ijkm} = 1 \quad \forall i \in I, j \in J \tag{3.15}$$

$$x_{ik} \leq M \cdot z_{ik}, \quad \forall i \in I, k \in H \tag{3.16}$$

$$x_{mj} \leq M \cdot z_{mj} \quad \forall m \in H, j \in J \tag{3.17}$$

$$x_{ik} \leq \bar{C}_{ik}[\theta_{ik}(1 - \alpha_{ik}) + \alpha_{ik}] \quad \forall i \in I, k \in H \tag{3.18}$$

$$x_{km} \leq \bar{C}_{km}[\theta_{km}(1 - \alpha_{km}) + \alpha_{km}] \quad \forall k \in H, m \in H \tag{3.19}$$

$$x_{mj} \leq \bar{C}_{mj}[\theta_{mj}(1 - \alpha_{mj}) + \alpha_{mj}] \quad \forall m \in H, j \in J \tag{3.20}$$

$$\sum_{i \in I} x_{ik} + \sum_{m \in H} x_{mk} \leq Ch_k \quad \forall k \in H \tag{3.21}$$

$$e(\tilde{T_{ik}}) = \bar{T}_{ik} + \beta \cdot \bar{T}_{ik} \cdot x_{ik}^{Nb} \cdot \frac{1 - {\theta_{ik}}^{1-Nb}}{\bar{C}_{ik}^{Nb}(1 - \theta_{ik})(1 - Nb)} \tag{3.22}$$

$$e(\tilde{T_{km}}) = \bar{T}_{km} + \beta \cdot \bar{T}_{km} \cdot x_{km}^{Nb} \cdot \frac{1 - {\theta_{km}}^{1-Nb}}{\bar{C}_{km}^{Nb}(1 - \theta_{km})(1 - Nb)} \tag{3.23}$$

$$e(\tilde{T_{mj}}) = \bar{T}_{mj} + \beta \cdot \bar{T}_{mj} \cdot x_{mj}^{Nb} \cdot \frac{1 - {\theta_{mj}}^{1-Nb}}{\bar{C}_{mj}^{Nb}(1 - \theta_{mj})(1 - Nb)} \tag{3.24}$$

$$\sum_{v\in V}\sum_{m\in H}(l''^{v}_{imj}) = W_{ij} \quad \forall\ i \in I, j \in J \tag{3.25}$$

$$\sum_{v\in V}(l^{v}_{ik}) = \sum_{v\in V}\sum_{m\in H}(l'^{v}_{ikm}) \quad \forall\ i \in I, k \in H \tag{3.26}$$

$$\sum_{v\in V}\sum_{k\in H}(l'^{v}_{ikm}) = \sum_{v\in V}\sum_{j\in J}(l''^{v}_{imj}) \quad \forall\ i \in I, m \in H \tag{3.27}$$

$$\sum_{i\in I}(l'^{v}_{ikm}) \leq Cap \quad \forall k, m \in H, v \in V \tag{3.28}$$

$$\sum_{i\in I}(l''^{v}_{imj}) \leq Cap \quad \forall m \in H, j \in J, v \in V \tag{3.29}$$

$$\sum_{v\in V}(l^{v}_{ik})/Cap = x_{ik} \quad \forall i \in I, k \in H \tag{3.30}$$

$$\sum_{v\in V}\sum_{i\in I}(l'^{v}_{ikm})/Cap = x_{km} \quad \forall\ k \in H, m \in H \tag{3.31}$$

$$\sum_{v\in V}\sum_{i\in I}(l''^{v}_{imj})/Cap = x_{mj} \quad \forall\ m \in H, j \in J \tag{3.32}$$

$$x_{ijkm} \in \{0,1\} \quad \forall i \in I, k \in H, m \in H, j \in J \tag{3.33}$$

$$z_{ik}, y_k, z_{mj} \in \{0,1\} \quad \forall i \in I, k \in H, m \in H, j \in J \tag{3.34}$$

$$b, b' \geq 0 \tag{3.35}$$

$$x_{ik}, x_{km}, x_{mj} \geq 0 \quad \forall i \in I, k \in H, m \in H, j \in J \tag{3.36}$$

$$l^{v}_{ik}, l'^{v}_{ikm}, l''^{v}_{imj} \geq 0 \quad \forall i \in I, k \in H, m \in H, j \in J, v \in V \tag{3.37}$$

$$e(\tilde{T}_{ik}), e(\tilde{T}_{km}), e(\tilde{T}_{mj}) \geq 0, \quad \forall i \in I, k \in H, m \in H, j \in J \tag{3.38}$$

Objective function (3.4) minimizes the sum of economic, environmental, and social costs. Constraints (3.5) and (3.6) indicate that each retailer and each customer are allocated to a single hub, respectively. Constraints (3.7) and (3.8) imply that only established hubs can be allocated to retailers and customers. Constraint (3.9) declares exactly that P hubs have to be opened in the network.

Constraints (3.10)-(3.12) state that the difference between the maximum and minimum number of allocations to opened hubs is less than ∂. Constraint (3.13) asserts if z_{ik} is equal to zero, there is not any route between retailer i and all possible customers as j by using hub k. Constraint (3.14) declares that if z_{mj} equals zero, there is not any route between customer j and all possible retailers as $i \in I$ using hub m. Constraint (3.15) implies that there is one route between each retailer and customer consisting and at least one hub and at most two hubs. Constraints (3.16)-(3.17) states that there is no flow on links that are not activated by allocating retailers and customers to hubs.

Constraints (3.18)-(3.20) are capacity reliability constraints which are explained comprehensively Section (3.6). Constraint (3.21) expresses that the sum of arrival flow to each hub must be less than the capacity of the hub. Constraints (3.22)-(3.24) Calculate the expected travel time for links from retailers to hubs, hubs to hubs, and hubs to customers. We have presented the details of these calculations in Section (3.7).

Constraint (3.25) states that the total number of products originating from retailer i arriving at customer j through any hubs and any vehicle must be equal to the demand of customer j from retailer i. Constraints (3.26) and (3.27) illustrate the flow conservation in hub nodes. Constraint (3.26) specifies that the total number of products arriving at the first intermediate hub k through any vehicle must be equal to the total number of products leaving the same hub by any vehicle. Constraint (3.27) states similar flow conservation for the second intermediate hub m. The remaining constraints pertain to the decision variables for loading and unloading. Constraint (3.28) states that the total number of products from various retailers loaded onto vehicle v traveling along link km can not exceed the vehicle's capacity. Similarly, constraint (3.29) specifies that the total number of products from various retailers loaded onto vehicle v traveling along link mj must respect the vehicle's capacity.

Constraints (3.30)-(3.32) illustrate the connection between the flow of vehicles on each link and the quantity of products carried by all vehicles traveling through that link. Constraint (3.30) states that dividing the total number of products loaded onto vehicles traveling in link ik by the vehicle capacity equals the flow of vehicles in link ik. Constraint (3.31) and (3.32) present similar calculations for links (k,m) and (m,j).Constraints (3.33)-(3.38) declare the decision variables.

3.6 Link Capacity Reliability Constraint

According to the introduction on link capacity disruptions, there is a certain probability that traffic flow on a link will exceed its capacity. It is important to consider the probability that ensures the traffic flow on network links remains at or below their capacities. In this context, capacity reliability is defined as the probability that traffic flow on a link exceeds its capacity with probability α_{ik} [28].

$$P\left\{x_{ik} \geq \tilde{C}_{ik}\right\} \leq \alpha_{ik} \tag{3.39}$$

The designed capacity of each link can be different from another, where $\tilde{C}_{ik}$ is a random variable defined by a specific probability density function. By considering a cumulative distribution function (CDF), the left-hand-side of equation **??** can be reformulated as **??**:

$$F_{\tilde{C}_{ik}}(x_{ik}) = P\left\{x_{ik} \geq \tilde{C}_{ik}\right\} \tag{3.40}$$

Taking into account equations 3.39 and 3.40, inequality 3.41 is derived.

$$F_{\tilde{C}_{ik}}(x_{ik}) \leq \alpha_{ik} \tag{3.41}$$

As the cumulative distribution functions (CDFs) are monotonic one-to-one functions, the inverse of inequality 3.41 can be expressed as:

$$x_{ik} \leq F^{-1}_{\tilde{C}_{ik}}(\alpha_{ik}) \tag{3.42}$$

To establish inequality 3.42, it is necessary to define an acceptable excessive probability α_{ik} and a specific cumulative distribution function (CDF) of link capacity $\tilde{C}_{ik}$. In this thesis, it is assumed that the link capacity follows a Uniform distribution, with the designed capacity serving as the upper bound ($\bar{C}_{ik}$) and the worst-degraded capacity ($\theta_{ik} \cdot \bar{C}_{ik}$) as the lower bound. Here, the worst-degraded capacity is represented by a fraction θ_{ik} of the designed capacity. The inverse CDF of the random variable $\tilde{C}_{ik}$ with a Uniform distribution [28] can be expressed as:

$$F^{-1}_{\tilde{C}_{ik}}(\alpha_{ik}) = \theta_{ik}\bar{C}_{ik} + \alpha_{ik}\bar{C}_{ik}(1-\theta_{ik}) = \bar{C}_{ik}[\theta_{ik}(1-\alpha_{ik}) + \alpha_{ik}] \tag{3.43}$$

The value of $\bar{C}_{ik}$ is deterministic and serves as the designed capacity of link (i,k). By employing equations 3.42 and 3.43, equation 3.44 can be utilized to establish the link capacity constraint.

$$x_{ik} \leq \bar{C}_{ik}[\theta_{ik}(1-\alpha_{ik}) + \alpha_{ik}] \tag{3.44}$$

3.7 Flow-dependent Link Travel Time Constraint

In this section, we introduce the BPR function link performance function, as presented in reference [28], denoted by equation 3.45, which has been tailored to suit our specific problem.

$$\tilde{T}_{ik}\left(x_{ik}, \tilde{C}_{ik}\right) = \bar{T}_{ik}\left[1 + \beta\left(\frac{x_{ik}}{\tilde{C}_{ik}}\right)^{Nb}\right] \tag{3.45}$$

Since $\tilde{C}ik$ is a random variable, $\tilde{T}ik$ is also a random variable. Using equation (3.45), the mean of $\tilde{T}_{ik}$ can be computed using equations (3.46). It is important to note that we use two notations, a lowercase letter (e) and an uppercase letter (E), to represent expectation. The lowercase letter signifies that the expectation includes a decision variable, making it a decision variable itself. On the other hand, the uppercase letter denotes a parameter.

$$e(\tilde{T_{ik}}) \quad = \quad e(\bar{T}_{ik}) + \beta \cdot \bar{T}_{ik} \cdot e[(\frac{x_{ik}}{\tilde{C_{ik}}})^{Nb}] \tag{3.46}$$

Under the assumption that $\tilde{C}ik$ is independent of the traffic flow amount (xik), and $\bar{T}_{ik}$ is a deterministic parameter, we have $E(\bar{T}_{ik}) = \bar{T}_{ik}$. Equations (3.46) can be reformulated as:

$$e(\tilde{T_{ik}}) = \bar{T}_{ik} + \beta \cdot \bar{T}_{ik} \cdot {x_{ik}}^{Nb} \cdot E[(\frac{1}{\tilde{C_{ik}}})^{Nb}] \tag{3.47}$$

The average of $\frac{1}{\tilde{C}_{ik}^{Nb}}$ are obtained as equations 3.48 by considering a Uniform distribution interval ($\theta_{ik}\bar{C}_{ik}$, $\bar{C}_{ik}$) for the link capacity random variable.

$$E(\frac{1}{\tilde{C_{ik}}^{Nb}}) \quad = \quad \int_{\theta_{ik}\bar{C}_{ik}}^{\bar{C}_{ik}} \frac{1}{\tilde{C_{ik}}^{Nb}} \frac{1}{\left(\bar{C}_{ik} - \theta_{ik}\bar{C}_{ik}\right)} d\tilde{C_{ik}} = \frac{1 - {\theta_{ik}}^{1-Nb}}{\bar{C}_{ik}^{Nb}(1-\theta_{ik})(1-Nb)} \tag{3.48}$$

The expression $\frac{1}{\bar{C}_{ik} - \theta_{ik}\bar{C}_{ik}}$ denotes the probability density function of the Uniform distribution, where the upper bound is $\bar{C}_{ik}$ and the lower bound is $\theta_{ik}\bar{C}_{ik}$. The derivation presented above is applicable for the general case when $Nb \neq 1$. However, when $Nb = 1$, the same approach is utilized. In this instance, the integration involves $\frac{1}{\tilde{C}_{ik}}$ within the algorithm function. Thus, we obtain:

$$e(\tilde{T_{ik}}) \quad = \quad \bar{T}_{ik} + \beta \cdot \bar{T}_{ik} \cdot x_{ik}^{Nb} \cdot \frac{1 - {\theta_{ik}}^{1-Nb}}{\bar{C}_{ik}^{Nb}(1-\theta_{ik})(1-Nb)} \tag{3.49}$$

Equations 3.49 indicate that for a specifically designed capacity $\bar{C}_{ik}$, the mean increases with traffic flow x_{ik}. This discussion holds for any distribution of $\tilde{C}_{ik}$. The total travel time for

the route is obtained by summing the link travel times, as demonstrated by:

$$\tilde{T}_{ijkm} = \tilde{T}_{ik} + \tilde{T}_{km} + \tilde{T}_{mj} \tag{3.50}$$

The expected travel time for a route is equal to the sum of the expected travel times for each link that makes up the route.

$$e(\tilde{T}_{ijkm}) \quad = \quad e(\tilde{T}_{ik}) + e(\tilde{T}_{km}) + e(\tilde{T}_{mj}) \tag{3.51}$$

Taking into account the aforementioned calculations 3.49 and 3.51, we derive the mean travel time for a route using equations 3.52:

$$\begin{aligned} e(\tilde{T}_{ijkm}) \quad &= \quad \bar{T}_{ik} + \beta \cdot \bar{T}_{ik} \cdot x_{ik}^{Nb} \cdot \frac{1 - {\theta_{ik}}^{1-Nb}}{\bar{C}_{ik}^{Nb}(1-\theta_{ik})(1-Nb)} \\ &+ \quad \bar{T}_{km} + \beta \cdot \bar{T}_{km} \cdot x_{km}^{Nb} \cdot \frac{1 - {\theta_{km}}^{1-Nb}}{\bar{C}_{km}^{Nb}(1-\theta_{km})(1-Nb)} \\ &+ \quad \bar{T}_{mj} + \beta \cdot \bar{T}_{mj} \cdot x_{mj}^{Nb} \cdot \frac{1 - {\theta_{mj}}^{1-Nb}}{\bar{c}_{mj}^{Nb}(1-\theta_{mj})(1-Nb)} \end{aligned} \tag{3.52}$$

Motivation for using BPR function

The BPR function was originally developed to model the relationship between travel time and traffic flow in urban road networks. While its initial purpose was focused on road traffic dynamics, The BPR function's mathematical structure models the relationship between flow, capacity, and travel time. It provides a flexible framework that is adaptable to other domains. This study focuses on a city logistics operator optimizing vehicle flows between origins and destinations with intermediate hub consolidation. Nodes represent origin and destination points, while transshipment edges symbolize virtual connections, such as shortest-path sub-networks. In this thesis, we leverage the BPR function to approximate flow-dependent delays in hubs and links within the PI framework.

One primary factor demonstrating the suitability of the BPR function for the supply chain context is its capacity to model delays caused by congestion in hubs and links. Similar to its application in capturing the impact of increasing traffic on road travel times, the BPR function effectively represents how congestion within hubs and links depends on the volume of goods processed relative to their capacities ([29], [30], [31]). Some studies examine the application of the BPR function within supply chain contexts using virtual links instead of actual roads.

- Karimi-Mamaghan et al. [29] address congestion in hub-and-spoke networks by considering delays resulting from the accumulation of flows in hubs and links within a logistics network, which significantly affects transportation times and delivery performance. They employ the BPR function to model delays on network links and hubs. Although their model does not involve a real-world case study, it effectively abstracts the congestion effects caused by flow accumulation in a logistics network. The use of the BPR function in this context demonstrates its flexibility in representing flow-dependent travel times in supply chains. This abstraction directly aligns with its application in virtual hubs and links, where congestion reflects processing times and flow volumes rather than urban traffic.

- Resat et al. [31] use real-world data to model inter-modal logistics, emphasizing flow-dependent delays in a multi-modal supply chain (road, rail, and sea). The BPR function is applied to estimate delays in transportation links caused by congestion due to freight movement. While the study uses real-world routes, the BPR function's application extends beyond actual roads. Its role in capturing flow-capacity interactions is conceptually aligned with its use in modeling virtual links in supply chain networks. The study demonstrates that the BPR function can effectively model delays in abstract logistics contexts, supporting its use in a virtual PI network.

- Mohammadi et al. [30] study models disruptions and uncertainties in a hub-and-spoke network using the BPR function to estimate delays on links. While the case study involves real locations and operational data, the application of the BPR function focuses on flow-capacity relationships, emphasizing supply chain dynamics rather than urban traffic conditions. The paper highlights the BPR function's capability to model congestion in links where delays result from disruptions or stochastic capacities. This aligns with its use in virtual links and hubs to capture delays caused by flow imbalances, reinforcing its flexibility beyond real-world road contexts.

Conclusion

- Beyond Real Roads: All three studies demonstrate the BPR function's ability to model congestion effects in logistics systems, focusing on flow-capacity relationships in the supply chain rather than actual urban traffic is supports its use in virtual or abstract logistics links.

- Real Data for Validation: The use of case studies with real data (Turkey and France, [31], [30]) validates the adaptability of the BPR function for practical applications.

However, the BPR functionality extends to abstract networks where traffic is driven by supply chain demands, not real urban traffic.

- Alignment with Supply Chain Dynamics: The studies illustrate how the BPR function can represent congestion within supply chains, providing a robust foundation for its use in modeling virtual links and hubs in the supply chain network.

CHAPTER 4 SOLUTION METHODOLOGY

Our proposed Mixed-Integer Nonlinear Programming (MINLP) model is optimally solved for only small instances by using the BARON Solver in the GAMS Platform. Therefore, we propose a GA to solve real-world problems efficiently. In the following, we first explain the solution representation in section 4.1 and then provide the details of the GA in section 4.2.

4.1 The Representation of the Solution

In our proposed GA, the solution representation encompasses hub locations (y_k), the allocation of non-hub nodes to hubs (z_{ik}, z_{km}), and route selection between all retailers and customers (x_{ijkm}). We use a vector of (n) elements for this purpose, where n represents the total number of nodes. In the following, we explain the content of this vector and how it represents a solution. While explaining, we include a numerical example i ∈ [1, 2, 3] as the set of retailers, j ∈ [4, 5, 6, 7] as the set of customers, and k, m ∈ [8, 9, 10] as the set of hubs.

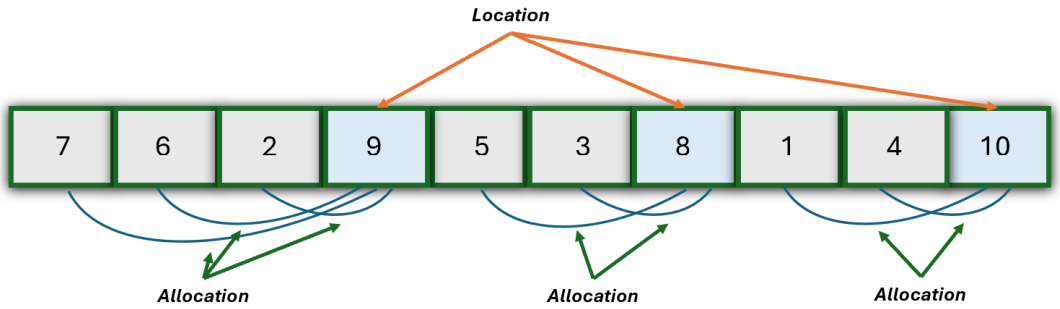

Figure 4.1 Chromosome Representation, z_{ik}, z_{mj} and y_k

- Location Decision Variable (y_k)
  For determining the hub cells in the chromosome, we randomly select ($P-1$) cells, except the last cell, for establishing hubs. The last cell is always designated as a hub. Then, we fill them with node indices randomly associated with hubs, ensuring these indices are unique and non-repeatable. For example, in Figure 4.1, the values 9, 8, and 10 in the location section represent the nodes established as hubs.

- Allocation Decision Variable (z_{ik}, z_{mj})

In the allocation process, each retailer or customer is assigned to the closest hub on its right-hand side. As an example, In Figure 4.1, non-hub nodes 7, 6, and 2 are retailers or customers that are allocated to hub 9. Also, non-hub nodes 5 and 3 are allocated to hub 8 and finally, non-hub nodes 1 and 4 are allocated to hub 10.

- Route Selection Decision Variable (x_{ijkm})
  In the proposed mathematical model, there are two possible routes: either route consists of a single hub or a route involves two hubs. Below, we separately define how the presented solution methodology addresses these two possible routes.

  - One-Hub
    In one-hub routes, there is only one hub connecting a retailer and a customer, serving as an intermediary point for consolidating and redistributing goods. These routes are constructed by grouping customers and retailers that are allocated to the same hub. The hub acts as a centralized location where goods from the retailer are collected and then distributed to the respective customers, optimizing transportation efficiency and reducing overall travel distances.
    We have provided an example in Figure 4.2. In this figure, In box 1, and 2 represent a retailer, and 6,7 represent customers, all of which are allocated to hub 9. Thus, one-hub routes 2-9-6 and 2-9-7 are created. The same logic applies to box 2 and box 3, with the routes displayed below each box.

  - Two-Hub
    In our solution representation, two-hub routes are formed by considering scenarios where the retailer and the customer are allocated to hubs in different boxes. This approach assumes a fully connected hub network, enabling seamless transfers between hubs. As a result, the route consists of two hubs: one connecting the retailer to the network and another linking the customer.
    Figures 4.3, 4.4 and 4.5 illustrate all possible combinations of two different cells. To explain the logic, we use Figure 4.3 for box 1 and box 2. In these boxes, cells 2,3 are retailers, and cells 5,6,7 are customers. Depending on their allocation to either hub 9 or hub 8, we have the following possible two-hub routes: 2-9-8-5, 3-8-9-6, and 3-8-9-7. For the other possible combinations of boxes, Figures 4.4 and 4.5 illustrate all possible two-hub routes using the same logic. In these figures, indices 1 and 4 represent a retailer and a customer, respectively.

- Other Decision Variables
  After defining the principal decision variables y_k, z_{ik}, z_{mj}, and x_{ijkm} by the given solution presentation, we can fix the values of the remaining decision variables as follows:
  - First, we set all variables l^v_{ik}, l'^v_{ikm} and l''^v_{imj} equal to 0.
  - If retailer i and customer j are connected by a single hub k in a one-hub route, we set $\sum_{v\in V} l''^v_{ikj} = W_{ij}$ and $\sum_{v\in V} l'^v_{ik} = W_{ij}$.
  - If retailer i and customer j are connected by two hubs k and m in a two-hub route, we set $l^v_{ik} = W_{ij}$, $\sum_{v\in V} l'^v_{ikm} += W_{ij}$ and $\sum_{v\in V} l''^v_{imj} = W_{ij}$.
  - Then, we set $l'^v_{ikm} = (\sum_{v\in V} l'^v_{ikm})/|V|$ and $l''^v_{imj} = (\sum_{v\in V} l''^v_{imj})/|V|$ to have the best chance to satisfy constraints (3.28) and (3.29).
  - After obtaining the values of loading/unloading variables, we can fix the values of x_{ik}, x_{km}, x_{mj} through constraints (3.30)-(3.32).
  - Finally, we set the values of $e(\tilde{T_{ik}})$, $e(\tilde{T_{km}})$, $e(\tilde{T_{mj}})$ through equations (3.22)-(3.24).

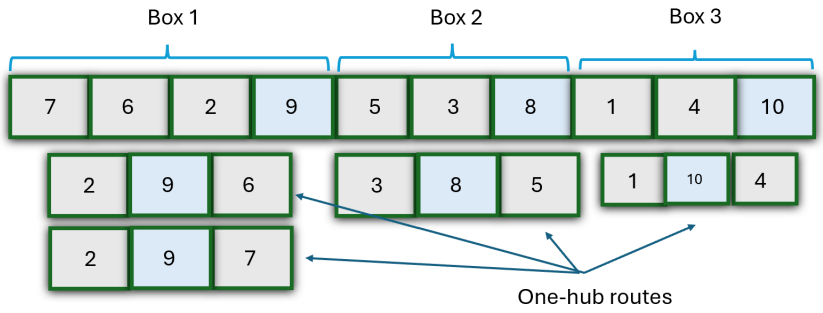

Figure 4.2 One-hub Routes between Box 1 and Box 2, x_{ijkm}

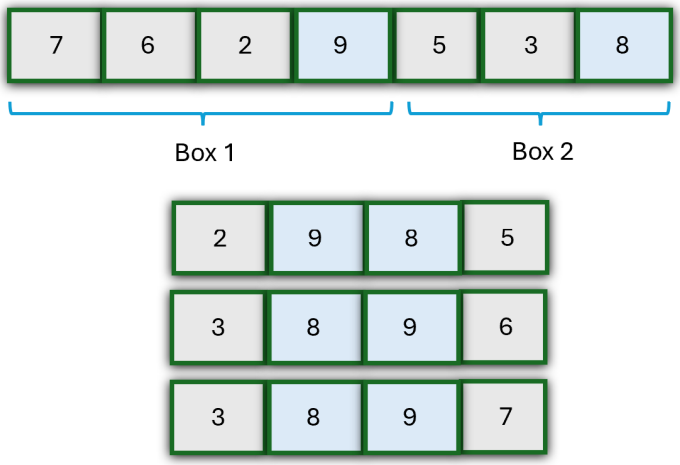

Figure 4.3 Two-hub routes Between Box 1 and Box 2, x_{ijkm}

4.2 Genetic Algorithm, (GA)

GA proposed by Holland [32] is a meta-heuristic algorithm inspired by natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are more likely to be selected for reproduction to produce offspring of the next generation. The fundamental goal of GA is to find the best solution or a sufficiently good solution to a problem in a large search space. GA is particularly useful for solving optimization and search problems. The main steps of GA are as follows:

1. Initialization: The initialization in a GA is the process of generating some initial solutions to the problem. Each individual in the population is called a chromosome, which is essentially a specific solution. Here, we show how we fill both location and allocation sections in each chromosome. The bits $(n+1$ to $n+p)$ are filled with random integer numbers within 1 to n for hub locations, while the $(n+p)^{th}$ bit is assigned the value n. These bits indicate the positions of the cells chosen as hubs in the allocation section. Then, within the permissible range of indices for hubs, we randomly select p indices and assign them to the positions specified in the allocation section for hubs. As an example in Figure 4.6, bits in positions 11 and 12 are randomly filled with integers between 1 to 9, while the 13^{th} bit is set to 10. Therefore, bits 11, 12, and 13 are filled with 4, 7, and 10. This means that the 4th, 7th, and 10th cells in the allocation section are designated as hub cells. Then, within the permissible range of indices for hubs, we generate 3 random numbers and place them in the 4^{th}, 7^{th}, and 10^{th} cells as hubs. In this example, we selected 9, 8, and 10 as hubs, and located them in the 4^{th}, 7^{th}, and 10^{th} cells.

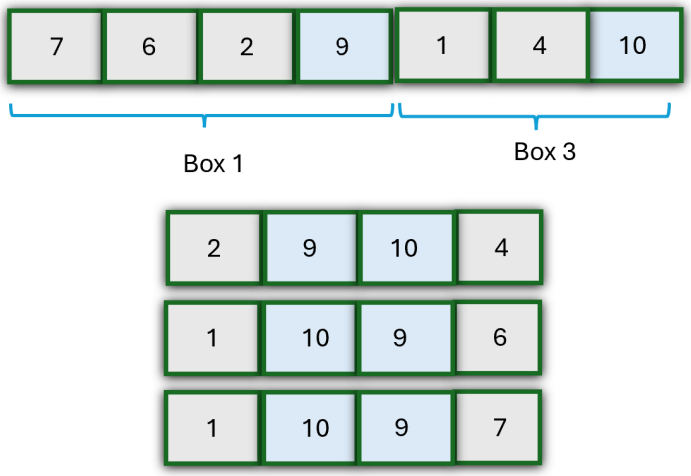

Figure 4.4 Two-hub Routes between Box 1 and Box 3, x_{ijkm}

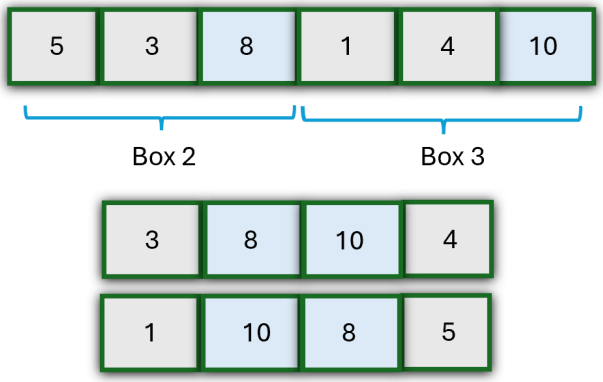

Figure 4.5 Two-hub routes Between Box 2 and Box 3, x_{ijkm}

As shown in Figure 4.7, the first n bits, except those allocated for hub locations, are populated with random numbers between zero and one. Then, they are sorted in descending order and filled with permissible indices for both retailers and customers. The approach promotes variation in the initial allocation choices, enabling the discovery of optimal allocations based on the randomly generated values. It establishes a foundation for subsequent optimization efforts. The proposed solution representation is continuous because the range of generated random numbers is both continuous and positive. This approach covers a broader search space than the discrete version and offers greater flexibility for adapting to specific problems. Furthermore, most heuristic and meta-heuristic algorithms are designed for continuous problems. That is why we decided to use this type of solution representation to develop our solution methodology [33].

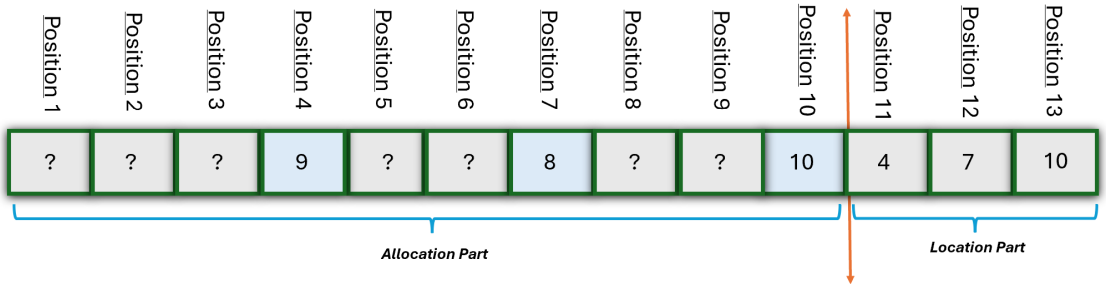

Figure 4.6 Initial Population Generation in GA, Location Part, y_k

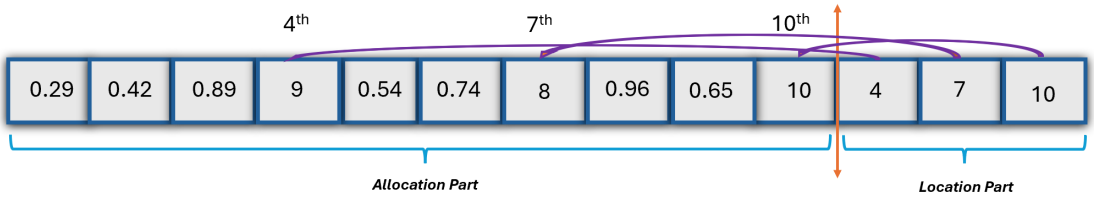

Figure 4.7 Initial Population Generation in GA, Allocation Part, z_{ik} and z_{mj}

2. Evaluation: In this step, all solutions are evaluated to determine how good they are. The objective values of solutions are used in the selection steps to guide the evaluation towards improved solutions. 3. Selection: Selection is the process of choosing which individuals from the current population will contribute to the next generation. Selection is based on the objective value of solutions, with stronger individuals, with fitter individuals being more likely to be selected. We use a roulette wheel selection for the selection process. While using the roulette wheel algorithm, we apply the elitism technique to ensure that the best existing solutions in the population are always transferred to the next generation. 4. Crossover: Crossover is a search operator in GA that combines two existing "parent" solutions to generate two new solutions, referred to as offspring. In our proposed GA algorithm, we used a single-point ordered crossover operator that generates new solutions as depicted in Figures 4.8 and 4.9. In the single-point ordered crossover technique, two-parent solutions (chromosomes) are selected from the population, and a random crossover point is chosen within the chromosome length. The first offspring is created by copying genes from one parent up to the crossover point, while the remainder is filled with non-duplicated genes from the second parent, preserving their order of appearance. This results in offspring that blend genetic material from both parents, maintaining the sequence from the first parent while incorporating non-duplicate genes from the second without any repetitions. There are various crossover techniques for discrete solutions, such as Uniform crossover, Arithmetic crossover, and Multi-Point crossover.

In the following explanation, we will explore two examples of single-point ordered crossover Figures 4.8 and 4.9, detailing how offspring are created by combining segments from two-parent solutions while maintaining the order of genes and avoiding duplicates. In the Figure 4.8 single-point ordered crossover example, a crossover point is chosen after the third gene, dividing the parents into segments. From Parent 1 (7, 6, 2), the first segment is selected for the offspring, while the remaining genes come from Parent 2 (3, 8, 1, 4, 9, 5, 10), ensuring no duplicates. The offspring starts with 7, 6, and 2 from Parent 1, followed by the next available non-duplicated genes from Parent 2 in their original order: 3, 8, 1, 4, 9, 5, and 10 leading to the final offspring: 7, 6, 2, 3, 8, 1, 4, 9, 5, 10. This process effectively combines genetic material from both parents while preserving the sequence and integrity of the genes. Moreover, in the Figure 4.9 single-point ordered crossover example, a crossover point is chosen after the sixth gene, dividing the parents into segments. From Parent 2 (i.e., 6, 3, 8, 1, 4, 9), the first segment is selected for the offspring, while the second segment is filled with genes from Parent 1 (i.e. 7, 2, 5, 10), ensuring no duplicates and maintaining the order. The offspring begins with 6, 3, 8, 1, 4, and 9 from Parent 2, followed by adding non-duplicated genes from Parent 1 in their original order: 7, 2, 5, and 10. This results in the final offspring:

6, 3, 8, 1, 4, 9, 7, 2, 5, 10, effectively blending the genetic material of both parents while preserving their sequence.

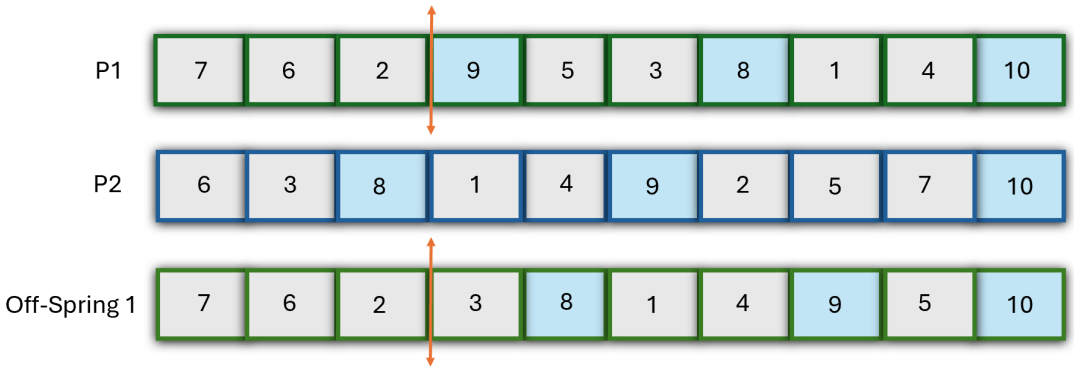

Figure 4.8 Ordered Crossover, Offspring 1

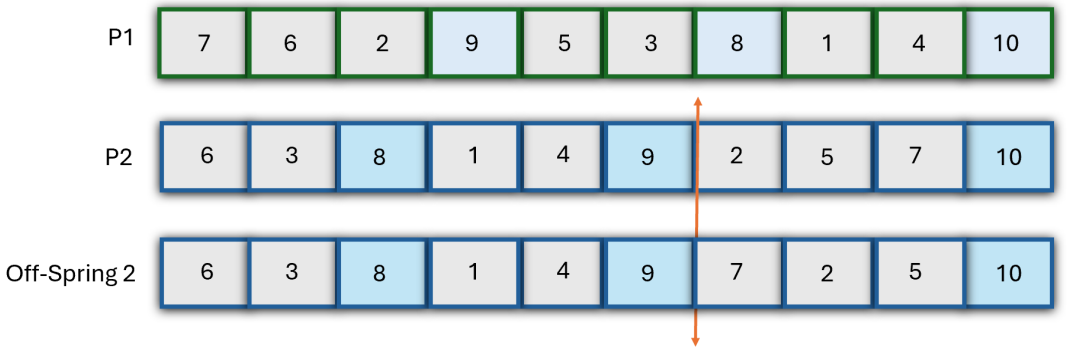

Figure 4.9 Ordered Crossover, Offspring 2

5. Mutation: Mutation introduces random changes to the chromosomes to maintain genetic diversity within the population. In the presented GA, the mutation process begins by defining a mutation rate, p, which determines the probability of a mutation occurring for an individual solution. This value is typically small, ranging from 0.01 to 0.05. For each individual in the population, a random number is generated, and if this number is less than the mutation rate p, the individual undergoes mutation. We use inversion mutation, which involves selecting two indices and reversing the order of elements between them. After performing Step 5, we check the termination criteria including the time limit and convergence of solutions and if they are not met we continue the algorithm from Step 2.

Starting with the original array [7,6,2,9,5,3,8,1,4,10], we proceed to perform an inversion mutation, an example of the mutation process explained earlier by randomly selecting the indices i=2 and j=6 (using 0-based indexing), which means we will invert the elements between these positions. The subset of the array identified between indices 2 and 6 is [2,9,5,3,8]. By reversing this subset of the array, we obtain the inverted subset of the array [8,3,5,9,2]. We then replace the original subset of the array in the array with the inverted version, .resulting in the transformation from [7, 6, 2, 9, 5, 3, 8, 1, 4, 10] to [7, 6, 8, 3, 5, 9, 2, 1, 4, 10] after the inversion, which is shown in Figure 4.2. 8. Elitism: Elitism is a technique used to ensure

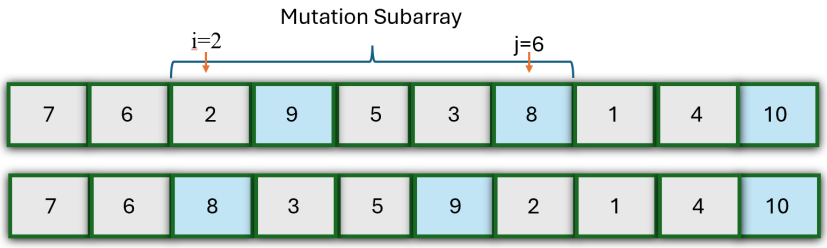

Figure 4.10 Ordered Crossover, OffSpring 1

that the fittest individuals are always preserved from one generation to the next. It prevents the loss of the best solutions found so far.

We solve the problem for both small size and big size with the proposed GA. In Chapter 5 we discuss comprehensively the obtained results by GA and Gams.

4.2.1 Incorporating the Chance-Constraint into a GA with a Penalty Approach

Constraints (3.18)-(3.20) are derived from the chance constraint (3.39). The chance constraint ensures that the probability of flow on a link exceeding its capacity stays below a predefined threshold α. This constraint minimizes the likelihood of capacity violations, thereby maintaining the reliability and feasibility of the solution. Integrating constraints (3.18)-(3.20) into the GA effectively incorporates the chance constraint into the optimization process. To guide the algorithm toward feasible solutions, a penalty-based approach is employed. The fitness function is augmented with a penalty term that quantifies the degree of violation of the right-hand side of constraints (3.18)-(3.20):

$$\text{Penalty} = \lambda \cdot \max\left(0, x_{ik} - \overline{C}_{ik}\left[\theta_{ik}(1-\alpha_{ik}) + \alpha_{ik}\right]\right),$$

where λ represents the penalty coefficient. The penalized fitness function is then defined as below. where $f(x)$ is the original fitness value of the solution.

$$f'(x) = f(x) + \lambda \cdot \text{Penalty}(x),$$

For each solution, the flow on all links is evaluated, and the right-hand side of constraints (3.18)-(3.20) is computed. If a violation occurs, a penalty is added to the fitness value, making the solution less likely to be selected during the GA's evolution process. Over successive generations, the penalty's influence is dynamically adjusted to allow initial exploration of the solution space while progressively enforcing stricter adherence to the constraints. This approach ensures convergence toward solutions that satisfy the chance constraints while optimizing the objective function.

To effectively balance constraint enforcement and objective optimization within the GA, the penalty coefficient λ was meticulously configured for each of the three instances with sizes 50, 100, and 150. For the smallest instance size 50, a moderate λ value of 10 was selected based on initial scaling analyses, ensuring that constraint violations would significantly impact the fitness without excessively restricting the search space. As the problem complexity increased with the larger instances, λ was correspondingly adjusted to 50 for the size 100 instance and 100 for the size 150 instance. This escalation accounts for the heightened likelihood of constraint violations and the increased number of flow variables that need to be regulated.

Additionally, an adaptive penalty adjustment strategy was employed across all instances, where λ was dynamically increased during successive generations if the proportion of feasible solutions fell below a predefined threshold (i.e., 80%). The 80% threshold means that at least 80% of the solutions in the population need to satisfy the constraints to maintain feasibility as a priority. Specifically, each time the feasible solution ratio falls below the 80% threshold, λ is incremented by multiplied by a scaling factor (i.e., $\lambda_{new} = \lambda_{current} * 1.5$) This approach allowed the metaheuristic algorithms to maintain a robust exploration of the solution space in the early stages while progressively enforcing stricter adherence to the chance constraints as the population converged. By tailoring λ to the specific scale and complexity of each instance, the algorithm ensured convergence toward solutions that not only optimize the objective function but also reliably satisfy the imposed chance constraints.

4.2.2 GA Settings

To effectively solve the optimization problem, the GA settings were carefully chosen to balance exploration, exploitation, and computational efficiency. Population sizes of 50, 100, and 150 were used for simpler and more complex instances, respectively, to maintain diversity and solution quality. The roulette wheel selection method was employed, selecting 50% of the population as parents for crossover. This probabilistic approach assigns each solution a portion of a virtual wheel based on its objective value, giving better solutions a higher chance of being selected while preserving diversity. A crossover rate of 70% was implemented, ensuring that a significant portion of selected parents undergo crossover to produce offspring. Additionally, steady-state replacement involves replacing only a few individuals at a time, typically the least fit members, with new offspring. This method helps maintain high-quality solutions. A low mutation rate of 0.05 was applied to introduce random changes, helping to avoid local optima. The GA was set to run for a maximum of 100 generations, balancing computational efficiency with solution accuracy. These settings enabled the GA to effectively explore the solution space, maintain population diversity, and deliver robust, high-quality results.

CHAPTER 5 RESULTS AND COMPUTATIONAL ANALYSIS

Here, we present some computational results on the model. We used the BARON Solver in the GAMS Platform to solve the MINP mathematical models. We ran all numerical experiments on a computer with two Intel Core i7-8565U CPUs running at 2001 Mhz. We considered a time limit of 4 hours.

5.1 Instance Generation

We categorize the instances into two groups of small and large instances, as illustrated in Table 5.1 and Table 5.2, respectively. Small instances have some nodes ranging from 6 to 10, while large instances have between 50 and 150 nodes. We created various instances by combining different numbers of retailers, hubs, and customers.

In all these instances, we have the number of retailers, hubs, and customers as shown in Tables 5.1 and 5.2. Regarding the traffic parameters, we set Nb and β to 0.15 and 4, respectively, as commonly used in the literature [34]. The fixed setup cost, F_k, is generated using a uniform distribution within the range [400, 600]. Transportation costs, including Fv_{ik}, Fv_{km}, and Fv_{mj}, are also drawn from a uniform distribution within the range [20, 40]. Demand values, W_{ij}, are generated from a uniform distribution within the range [400, 700]. The exceed probability values, α_{ik}, α_{km}, and α_{mj}, are sampled from a uniform distribution in the range [0.15, 0.3]. For free travel time, $\bar{T}_{ik}$, $\bar{T}_{km}$, and $\bar{T}_{mj}$, we use a uniform distribution within the range [100, 300]. Nominal capacities, $\bar{C}_{ik}$, $\bar{C}_{km}$, and $\bar{C}_{mj}$, are drawn from a uniform distribution within the range [20000, 70000]. Finally, disruption multipliers, θ_{ik}, θ_{km}, and θ_{mj} are sampled from a uniform distribution within the range [0.15, 0.3]. In our model, the generated random parameters are assumed to be independent, meaning that the variability in one parameter does not influence or depend on the variability of the other.

The parameter generation described above is applied to both small and large instances. However, for the large instances, we specifically set the disruption multipliers, θ_{ik}, θ_{km}, and θ_{mj}, to assess algorithms, GA and ICA, performance under different levels of disruption: mild, moderate, and severe, with values of 0.9, 0.5, and 0.15, respectively.

5.2 Results

In the following computational experiments, we aim to answer the following questions:

- What are the limitations of commercial solvers in solving the proposed mathematical

model?

- How does the GA algorithm perform on both small and large instances?
- How does the GA perform under varying levels of disruption intensity, such as mild, moderate, and severe?

Initially, we compare the solutions obtained from the exact mathematical programming model with those from the GA for small-sized instances, as detailed in Table 5.1 for answering the questions 1 and 2. In this table, LB and UB stand for lower and upper bounds, respectively. Also, the optimality gap for the GA is computed using its upper bound and the lower bound of the proposed mathematical model. The results indicate that the GA successfully approximates many of the optimal solutions of the mathematical model, demonstrating its effectiveness in solving large instances where exact solutions may be computationally impractical. For instances where the GA did not reach the optimal solution, the optimality gap was less than 3%, suggesting that the GA is well-suited for small-sized problems by offering near-optimal solutions with competitive computational times. Specifically, the GA performs within 20 minutes, demonstrating superior efficiency compared to the exact method, although it does not guarantee optimal solutions.

The significant difference in the BARON's computational time for solving different numbers of nodes likely stems from the complexity of the branching process used in global optimization ([35], [36]). In terms of the complexity of the initial node, the initial node in BARON requires extensive computation to define problem bounds, set constraints, and estimate feasible regions. This process is particularly demanding for nonlinear problems with complex constraints, as BARON uses interval arithmetic and convex under-estimators to ensure global optimality. Significant solution space exploration may also be needed to establish feasible bounds, resulting in longer computational times ([35], [36]). In terms of branch-and-bound variability, the computational time for opening nodes in BARON's algorithm varies with constraint feasibility and branch complexity. Extensive calculations are required when solving the initial node with many constraints or tightly bounded variables, which increases computational time. However, once bounds are established, subsequent nodes often require fewer calculations, as BARON leverages these bounds to efficiently prune or skip branches, significantly reducing time per node ([35], [36]).

In terms of node structure and solution density, computational time per node varies based on constraint density and solution feasibility across the solution space. Dense feasible regions within a complex node may require more processing time compared to simpler nodes. If the initial node involves a challenging non-convex region, it demands detailed exploration, whereas other nodes in simpler regions can be processed more efficiently ([35], [36]). In

summary, the discrepancy in computational time per node occurs because the BARON's processing time isn't uniform; it depends on the complexity of each node's constraints, the initialization process, and the branching strategy. The initial node may represent a particularly challenging region in the solution space that requires extensive processing, while subsequent nodes benefit from established bounds and simpler constraints. Therefore, even the optimal solutions do not follow a specific pattern for computational time.

We could not solve the model using exact methods for more than 50 nodes due to a dramatic increase in computational time. Even we could not find the feasible solutions. Consequently, we employed the GA to address the larger instances, which are detailed in Table 5.2. In our thesis, we selected the Imperialist Competitive Algorithm (ICA) as a benchmark to compare the results of the GA solutions. ICA, a population-based metaheuristic inspired by imperialistic competition, models colonies as potential solutions that compete to achieve superior outcomes. By using ICA as a reference, we sought to evaluate the performance and effectiveness of the GA in finding high-quality solutions. This comparison provides valuable insights into the strengths and weaknesses of both algorithms, aiding in assessing the efficiency of the GA for addressing our problem.

ICA, as originally introduced in the 2007 IEEE paper [37], was developed to tackle global optimization challenges through a unique competitive framework. This evolutionary approach has since been applied to a wide range of continuous optimization problems, establishing its credibility and comparability to GA in optimization studies [37]. Given that our thesis addresses complex PI logistics optimization with uncertain travel times and capacity constraints, ICA's iterative competitive mechanism is especially relevant for large-scale applications [37]. Also, ICA's ability to balance exploration and exploitation makes it a valuable alternative in research for comparing algorithmic efficiency under varying problem complexities [37]. Furthermore, ICA is well-suited to various optimization tasks and is applied across diverse fields, including engineering, logistics, machine learning, and finance, showcasing its adaptability and effectiveness in addressing complex challenges [37]. These points collectively underscore ICA's credibility as a logical and effective benchmark and alternative to GA in our study, providing a comparison in high adaptability and complexity handling. The outline of ICA is provided in Appendix A. For the implementation of ICA, we use the same solution representation introduced in Section 4.1 In Table 5.2, we have $Delta_{(\mathrm{ICA,GA})}(\%) = 100 \cdot \frac{UB_{\mathrm{ICA}} - UB_{\mathrm{GA}}}{UB_{\mathrm{GA}}}$. where UB_{ICA} and UB_{GA} present the upper bounds of ICA and GA, respectively.

Our results indicate the upper bounds obtained by GA are superior to those from ICA, More specifically, GA finds solutions that are on average 15.98%, 22.66%, and 26.07% stronger than those of ICA for different values of the traffic parameter θ. Our results also show that GA requires more computational time due to the additional operations involved in crossover and

mutation. Both GA and ICA experience increased computational time and upper bounds with varying disruption intensities—mild, moderate, and severe. Another notable observation is that both algorithms are sensitive to the number of hubs: as the number of hubs increases, the problem becomes more complex and demands more computational time. Nevertheless, GA consistently outperforms ICA in these scenarios.

5.3 Value of Solution Improvement Test (VSIT)

The Value of Solution Improvement Test (VSIT) measures the relative deviation of the simplified objective value compared to the original objective value reported for large instances solved by the GA in Table 5.2. The formula for VSIT is given as:

$$\text{VSIT } (\%) = 100 \times \frac{\text{Simplified Objective Value} - \text{Original Objective Value}}{\text{Original Objective Value}}$$

To calculate the simplified objective value, we first remove the uncertainty components from constraints (3.22) to (3.24) as below.

$$e(\tilde{T_{ik}}) = \bar{T}_{ik} \tag{5.1}$$

$$e(\tilde{T_{km}}) = \bar{T}_{km} \tag{5.2}$$

$$e(\tilde{T_{mj}}) = \bar{T}_{mj} \tag{5.3}$$

Next, we solve the simplified model using GA. After obtaining a solution, we fixed the taken uncertain parts with obtained solution (i.e, $x_{ik} = X_{ik}, x_{km} = X_{km}, x_{mj} = X_{mj}$).

$$e(\tilde{T_{ik}}) = \bar{T}_{ik} + \beta \cdot \bar{T}_{ik} \cdot X_{ik}^{Nb} \cdot \frac{1 - \theta_{ik}{}^{1-Nb}}{\bar{C}_{ik}^{Nb}(1 - \theta_{ik})(1 - Nb)} \tag{5.4}$$

$$e(\tilde{T_{km}}) = \bar{T}_{km} + \beta \cdot \bar{T}_{km} \cdot X_{km}^{Nb} \cdot \frac{1 - \theta_{km}{}^{1-Nb}}{\bar{C}_{km}^{Nb}(1 - \theta_{km})(1 - Nb)} \tag{5.5}$$

$$e(\tilde{T_{mj}}) = \bar{T}_{mj} + \beta \cdot \bar{T}_{mj} \cdot X_{mj}^{Nb} \cdot \frac{1 - \theta_{mj}{}^{1-Nb}}{\bar{C}_{mj}^{Nb}(1 - \theta_{mj})(1 - Nb)} \tag{5.6}$$

Finally, solve the problem with fixed uncertainty components in constraints (5.4), (5.5) and (5.6). We compute the objective function and call it a simplified objective value. The results are presented in Table 5.2, including a column with the computed VSIT values for different instances. These values represent the percentage deviation of the simplified objective value compared to the original objective value. In fact we substitute the uncertain parts of constraint (3.18), (3.19) and (3.20) with a constant value derived by solving the problem

with constraints (5.4), (5.5) and (5.6).

The VSIT results show the effects of considering uncertainty in the model. For instance, with 50 Nodes, 15 Retailers, 15 Hubs, 20 Customers and $\theta = 0.15$ the VSIT value is 23.28. This demonstrates that incorporating uncertainty into the mathematical model, as represented in equations (3.18), (3.19), and (3.20), adds meaningful value. It highlights the significance of properly accounting for uncertainty within the model. On the other hand, solving the problem using simplified equations without considering uncertainty results in a cost for neglecting these uncertain factors. Moreover, these results demonstrate the effectiveness of the GA in handling uncertainty as well. It represents how much GA improves the objective functions by considering uncertainty in the model. The inclusion of the VSIT(%) column in Table 5.2 provides a quantitative measure of the added value of considering uncertainty by GA.

5.4 Discussion

It is vital to consider the costly and strategic decisions in the PI transportation network in facing uncertainties. The impact of travel time uncertainty is a critical factor in designing PI networks, yet this aspect is often overlooked in the existing literature reviews. Travel time uncertainty can lead to delays and variability in the delivery of goods and services, impacting overall network performance, resource allocation, and cost-effectiveness. By accounting for this uncertainty, the network can be designed to be more resilient, adaptive, and capable of handling real-world complexities, ultimately improving service quality and operational efficiency. Our proposed model takes into account uncertain flow-dependent travel time in a PI network.

Our results indicate that while we can obtain optimal solutions for small instances with up to 10 nodes using a commercial solver, the GA exhibits superior performance in terms of computational efficiency. Additionally, the GA algorithm offers the advantage of quickly finding high-quality solutions for larger instances—up to 150 nodes and 60 hubs—under various disruption conditions (mild, moderate, and severe), with reasonable computation times of under an hour. This performance is notably better than ICA in both the quality of the upper bound found and the computational time required. Therefore, the proposed GA has the potential to be used by practitioners for daily operations and strategic scale decisions.

Table 5.1 Small Set of Instances, Exact Mathematical Programming Model and GA

Data				Exact Mathematical Programming Model					GA		
Node	Retailer	Hub	Customer	Number of Nodes	Time(sec)	LB	UB	Gap	Time(sec)	UB	Gap
6	2	2	2	1	1	86,908	86,908	0.0	22	86,908	0.00
7	2	2	3	83	9	97,542	97,542	0.0	728	97,542	0.00
	2	3	2	1	2	109,955	109,955	0.0	354	109,955	0.00
	3	2	2	1	2	96,244	96,244	0.0	280	96,244	0.00
8	2	2	4	1	2	118,794	118,794	0.0	496	118,794	0.00
	2	4	2	1	599	194,449	194,449	0.0	702	194,449	0.00
	4	2	2	1	2	124,719	124,719	0.0	481	124,719	0.00
	2	3	3	1	6	152,896	152,896	0.0	320	152,896	0.00
	3	2	3	1	1	118,836	118,836	0.0	173	118,836	0.00
	3	3	2	1	337	148,700	148,700	0.0	125	148,700	0.00
9	5	2	2	71	236	134,040	134,040	0.0	314	134,040	0.00
	2	5	2	1	1029	235,509	235,509	0.0	757	241,008	2.33
	2	2	5	1	1	145,315	145,315	0.0	312	145,315	0.00
	4	3	2	1	402	160,272	160,272	0.0	945	160,272	0.00
	3	4	2	1	3438	212,920	212,920	0.0	120	218,699	2.71
	2	3	4	1	285	163,730	163,730	0.0	145	163,730	0.00
10	6	2	2	1	38	161,009	161,009	0.0	300	161,009	0.00
	5	3	2	5	54	199,810	199,810	0.0	975	199,810	0.00
	3	5	2	1	15259	288,028	288,028	0.0	1050	291,018	1.04
	2	3	5	3	212	198,351	198,351	0.0	114	198,351	0.00
	5	2	3	75	472	161,526	161,526	0.0	317	161,526	0.00
	3	2	5	1	2	177,932	177,932	0.0	290	177,932	0.00
	2	5	3	3	3081	293,457	293,457	0.0	1200	299,047	1.90
	4	3	3	2	60	199,342	199,342	0.0	320	199,342	0.00

Table 5.2 Big set of Instances, GA and ICA

	Data				GA		ICA			
θ	Node	Retailer	Hub	Customer	$Time_{\mathrm{GA}}(sec)$	UB_{GA}	$Time_{\mathrm{ICA}}(sec)$	UB_{ICA}	$Delta_{(\mathrm{ICA,\ GA})}(\%)$	$VIST_{(\mathrm{GA})}(\%)$
0.15	50	15	15	20	1613	1,702,568	987	2,255,432	32.47	23.28
		15	20	15	1375	1,949,992	1380	2,975,212	52.58	17.26
		20	15	15	1199	1,720,356	913	2,271,347	32.03	26.05
	100	30	30	40	3209	3,019,181	4783	3,197,354	5.90	34.90
		30	40	30	5232	3,364,180	5124	3,765,734	11.94	8.32
		40	30	30	3509	3,148,150	4678	3,234,579	2.75	29.38
	150	45	45	60	10003	3,590,351	8564	3,565,739	-0.69	22.13
		45	60	45	8236	4,065,100	9945	4,043,621	-0.53	7.09
		60	45	45	8906	3,301,162	8645	3,543,648	7.35	23.78
Average					4809	2,873,449	5002	3,205,852	15.98	19.43
0.5	50	15	15	20	1533	1,411,304	1002	2,172,738	53.95	24.12
		15	20	15	1284	1,883,027	1402	2,617,352	39.00	19.88
		20	15	15	1546	1,428,194	783	2,181,649	52.76	21.55
	100	30	30	40	3021	2,909,998	4897	3,136,242	7.77	28.04
		30	40	30	4810	3,179,061	5345	3,673,542	15.55	6.57
		40	30	30	3101	3,050,750	5262	3,225,231	5.72	21.38
	150	45	45	60	9876	3,257,435	8977	3,565,739	9.46	24.80
		45	60	45	9966	3,564,739	10230	3,965,833	11.25	15.39
		60	45	45	7995	3,265,793	9034	3,543,648	8.51	23.22
Average					4792	2,661,145	5215	3,120,219	22.66	20.10
0.9	50	15	15	20	1348	1,301,215	987	2,110,791	62.22	24.41
		15	20	15	1145	1,826,028	1293	2,563,572	40.39	17.99
		20	15	15	1483	1,400,429	1522	2,104,538	50.28	29.83
	100	30	30	40	2912	2,713,030	4897	3,075,342	13.35	23.09
		30	40	30	4140	2,901,291	5219	3,783,545	30.41	12.05
		40	30	30	2941	2,843,171	4876	3,163,428	11.26	29.39
	150	45	45	60	9876	3,267,435	8345	3,517,354	7.65	21.70
		45	60	45	9966	3,564,739	9875	3,987,534	11.86	6.47
		60	45	45	7995	3,265,793	8278	3,501,754	7.23	21.73
Average					4645	2,564,792	5032	3,089,762	26.07	19.60

CHAPTER 6 CONCLUSION

This research explores the integration of PI principles to enhance sustainability across logistics and supply chain operations by addressing economic, environmental, and social dimensions under uncertain flow-dependent travel times. Our work emphasizes the integration of modular containers within PI networks to optimize resource use, reduce costs, and enhance environmental responsibility. Through a detailed analysis of the proposed mixed-integer nonlinear mathematical model using two meta-heuristic algorithms, the GA and the ICA, we demonstrated that the GA outperforms the ICA in approximating optimal solution.

6.1 Summary of Works

The thesis explores the application of the PI to enhance logistics and supply chain sustainability. By integrating modular containerization, the study illustrates how PI can reduce logistics costs, minimize environmental impact, and improve social welfare. The research addresses the challenges of flow-dependent travel times and loading/unloading processes. It offers a sophisticated mathematical model to navigate these complexities. The model incorporates uncertainties related to link capacity and travel times, advancing the field by presenting a more realistic and detailed approach to PI network optimization.
In our study, we tackled the problem for small instances with up to 10 nodes using the BARON Solver in the GAMS Platform, a GA, and an imperialist competitive algorithm. The BARON Solver in the GAMS Platform was highly effective for smaller cases, although its computational time increased significantly as the problem size as the problem size neared 10 nodes. For larger problems, the GA, while more time-consuming, provided better solutions. Nonetheless, the computational time difference between the GA and ICA was negligible for problems with up to 150 nodes.

6.2 Limitations

Despite its contributions, the research has several limitations, as follows:

- Assumptions and Simplifications: Certain assumptions, such as the specific form of the BPR function and the representation of uncertainties, may not fully capture real-world dynamics and variations in PI networks.
- Data Availability: Accurate and comprehensive data on travel times, link capacities,

and loading/unloading operations are essential for calibrating and validating the model. Ensuring data availability and quality in practice can be challenging.

- Future Demand Forecasting: The current model does not explicitly integrate future demand forecasting, which could enhance its accuracy and realism. Incorporating demand forecasts would allow for better anticipation of changes in logistics needs and more effective planning and resource allocation.

6.3 Future Research

To address these limitations and improve upon the current work, future research could focus on the following areas:

- Incorporation of Demand Forecasting: Integrate advanced demand forecasting techniques into the model to improve its accuracy and realism. This would allow for better anticipation of future logistics needs and more effective planning, improving overall model performance.
- Scalability and Efficiency: Investigate methods to simplify and optimize the model for larger and more complex PI networks. Techniques such as decomposition, heuristic approaches, or approximation methods could improve computational efficiency.
- Advanced Uncertainty Modeling: Develop more sophisticated models for capturing and managing various types of uncertainties, including dynamic traffic patterns, real-time disruptions, and other operational variables that impact PI networks.
- Real-World Validation: Conduct empirical studies or pilot projects to validate the model's effectiveness in practical PI implementations. Assessing performance in real-world scenarios will determine practical utility and identify areas for improvement. Utilizing benchmark examples enables rigorous evaluation of the proposed mathematical model by providing a common reference point for comparison.
- Technological Integration: Investigate the integration of emerging technologies, such as blockchain, IoT, and machine learning, to enhance data accuracy, real-time monitoring, and coordination within PI networks. This could improve the model's adaptability and responsiveness to dynamic conditions.

By addressing these limitations and exploring future research directions, the model can be further developed. This will provide a more accurate, scalable, and practical tool for optimizing PI networks and enhancing sustainability in logistics.

REFERENCES

[1] B. Montreuil, "Toward a physical internet: meeting the global logistics sustainability grand challenge," *Logistics Research*, vol. 3, pp. 71–87, 2011.

[2] H. Treiblmaier, K. Mirkovski, and P. B. Lowry, "Conceptualizing the physical internet: literature review, implications and directions for future research," in *11th CSCMP Annual European Research Seminar, Vienna, Austria, May*, 2016.

[3] B. Montreuil and B. Montreuil, "Physical internet manifesto, version 1.11. 1," *CIRRELT Interuniversity Research Center on Enterprise Networks, Logistics and Transportation*, pp. 2–3, 2012.

[4] E. Ballot, B. Montreuil, and R. D. Meller, *The physical internet*. La Documentation Française, 2014.

[5] H. Luo, S. Tian, and X. T. Kong, "Physical internet-enabled customised furniture delivery in the metropolitan areas: digitalisation, optimisation and case study," *International Journal of Production Research*, vol. 59, no. 7, pp. 2193–2217, 2021.

[6] X. Peng *et al.*, "Resilience planning for physical internet enabled hyperconnected production-inventory-distribution systems," *Computers & Industrial Engineering*, vol. 158, p. 107413, 2021.

[7] M. Li *et al.*, "A physical internet (pi) based inland container transportation problem with selective non-containerized shipping requests," *International Journal of Production Economics*, vol. 245, p. 108403, 2022.

[8] L. Zheng, P. Beem, and K.-H. G. Bae, "Assessment of the physical internet enabled urban logistics using agent-based simulation," *International Journal of Logistics Systems and Management*, vol. 33, no. 4, pp. 441–466, 2019.

[9] E. K. Leung, C. K. H. Lee, and Z. Ouyang, "From traditional warehouses to physical internet hubs: A digital twin-based inbound synchronization framework for pi-order management," *International Journal of Production Economics*, vol. 244, p. 108353, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925527321003297

[10] M. Lin *et al.*, "The value of the physical internet on the meals-on-wheels delivery system," *International Journal of Production Economics*, vol. 248, p. 108459, 2022.

[11] T. Chargui *et al.*, "Multi-objective cross-docking in physical internet hubs under arrival time uncertainty," in *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future: Proceedings of SOHOMA 2020*. Springer, 2021, pp. 460–472.

[12] F. Pan *et al.*, "Perishable product bundling with logistics uncertainty: Solution based on physical internet," *International Journal of Production Economics*, vol. 244, p. 108386, 2022.

[13] F. Essghaier *et al.*, "Fuzzy multi-objective truck scheduling in multi-modal rail–road physical internet hubs," *Computers & Industrial Engineering*, vol. 182, p. 109404, 2023.

[14] X. Liu *et al.*, "Multi-period stochastic logistic hub capacity planning for relay transportation," in *IISE Annual Conference Proceedings, Institute of Industrial and Systems Engineers (IISE)*, 2024.

[15] X. Liua *et al.*, "Dynamic hub capacity planning in hyperconnected transportation networks under uncertainty."

[16] U. Venkatadri, K. S. Krishna, and M. A. Ülkü, "On physical internet logistics: modeling the impact of consolidation on transportation and inventory costs," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 4, pp. 1517–1527, 2016.

[17] E. Puskás, Á. Budai, and G. Bohács, "Optimization of a physical internet based supply chain using reinforcement learning," *European Transport Research Review*, vol. 12, no. 1, pp. 1–15, 2020.

[18] M. Fazili *et al.*, "Physical internet, conventional and hybrid logistic systems: a routing optimisation-based comparison using the eastern canada road network case study," *International Journal of Production Research*, vol. 55, no. 9, pp. 2703–2730, 2017.

[19] A. Pal and K. Kant, "A food transportation framework for an efficient and worker-friendly fresh food physical internet," *Logistics*, vol. 1, no. 2, p. 10, 2017.

[20] S. S. Chadha, M. A. Ülkü, and U. Venkatadri, "Freight delivery in a physical internet supply chain: an applied optimisation model with peddling and shipment consolidation," *International Journal of Production Research*, vol. 60, no. 16, pp. 4995–5011, 2022.

[21] Y. Yang, S. Pan, and E. Ballot, "Innovative vendor-managed inventory strategy exploiting interconnected logistics services in the physical internet," *International Journal of Production Research*, vol. 55, no. 9, pp. 2685–2702, 2017.

[22] S.-f. Ji, X.-s. Peng, and R.-j. Luo, "An integrated model for the production-inventory-distribution problem in the physical internet," *International Journal of Production Research*, vol. 57, no. 4, pp. 1000–1017, 2019.

[23] X.-s. Peng, S.-f. Ji, and T.-t. Ji, "Promoting sustainability of the integrated production-inventory-distribution system through the physical internet," *International Journal of Production Research*, vol. 58, no. 22, pp. 6985–7004, 2020.

[24] C. Guo *et al.*, "An intelligent open trading system for on-demand delivery facilitated by deep q network based reinforcement learning," *International Journal of Production Research*, pp. 1–23, 2024.

[25] T. Chargui *et al.*, "Proposal of a multi-agent model for the sustainable truck scheduling and containers grouping problem in a road-rail physical internet hub," *International Journal of Production Research*, vol. 58, no. 18, pp. 5477–5501, 2020.

[26] B. Q. Tan *et al.*, "A reverse vickrey auction for physical internet (pi) enabled parking management systems," *International Journal of Production Economics*, vol. 235, p. 108083, 2021.

[27] N. Grover and B. Montreuil, "Dynamic time-based parcel consolidation and container loading in hyperconnected logistic hubs," in *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 2021, pp. 956–961.

[28] H. K. Lo, X. Luo, and B. W. Siu, "Degradable transport network: Travel time budget of travelers with heterogeneous risk aversion," *Transportation Research Part B: Methodological*, vol. 40, no. 9, pp. 792–806, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261505001177

[29] M. Karimi-Mamaghan *et al.*, "Hub-and-spoke network design under congestion: A learning based metaheuristic," *Transportation research part e: logistics and transportation review*, vol. 142, p. 102069, 2020.

[30] M. Mohammadi, P. Jula, and R. Tavakkoli-Moghaddam, "Reliable single-allocation hub location problem with disruptions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 123, pp. 90–120, 2019.

[31] H. G. Resat and M. Turkay, "A bi-objective model for design and analysis of sustainable intermodal transportation systems: a case study of turkey," *International Journal of Production Research*, vol. 57, no. 19, pp. 6146–6161, 2019.

[32] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[33] M. Sadeghi *et al.*, "A new stochastic approach for a reliable p-hub covering location problem," *Computers & Industrial Engineering*, vol. 90, pp. 371–380, 2015.

[34] H. Z. Aashtiani, "The multi-modal traffic assignment problem." Ph.D. dissertation, Massachusetts Institute of Technology, 1979.

[35] M. Tawarmalani and N. V. Sahinidis, "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study," *Mathematical programming*, vol. 99, no. 3, pp. 563–591, 2004.

[36] N. V. Sahinidis, "Baron: A general purpose global optimization software package," *Journal of global optimization*, vol. 8, pp. 201–205, 1996.

[37] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *2007 IEEE congress on evolutionary computation*. Ieee, 2007, pp. 4661–4667.

APPENDIX A

IMPERIALIST COMPETITIVE ALGORITHM, ICA

The Imperialist Competitive Algorithm (ICA), introduced by E. Atashpaz-Gargari [37], is a computational method inspired by the socio-political process of imperialism and imperialistic competition. It is utilized to solve various optimization problems by simulating the competition among empires for dominance. Understanding the key terminology within ICA is essential for grasping its mechanics:

- Country: Represents an individual solution within the search space. Analogous to chromosomes in GA, each country embodies a potential solution to the optimization problem.
- Empire: Consists of an imperialist (the leading country) and its colonies (other countries). Empires are the primary entities that compete within the algorithm.
- Imperialist: The dominant country within an empire, representing the best solution among its colonies. Imperialists guide the movement of colonies toward optimal solutions.
- Colony: Countries within an empire that are influenced by the imperialist. Colonies move toward the imperialist in the search space, simulating the assimilation process.
- Assimilation: The process by which colonies move closer to their imperialist in the search space, reflecting cultural or ideological assimilation. This movement is intended to exploit the search space around the imperialist.
- Revolution: Introduces random changes to some colonies, akin to mutations in GA. Revolution helps maintain diversity within the population and prevents premature convergence.
- Imperialistic Competition: A process where empires compete to take control of colonies from weaker empires. This competition drives the algorithm toward optimal solutions by eliminating weaker empires and expanding stronger ones.
- Total Cost: A metric that combines the cost of the imperialist and a portion of the costs of its colonies. It's used to evaluate the overall power of an empire.

- Colonial Takeover: Occurs when a colony's position improves beyond that of the imperialist, leading to a swap in roles. This mechanism ensures that better solutions are promoted within the empire.
- Elimination of Weak Empires: Empires that lose all their colonies through competition are removed from the process. This elimination focuses the search on more promising areas of the solution space.

These terms collectively describe the components and operations of ICA, which facilitate its function as an optimization tool mimicking the dynamics of imperialistic competition. The ICA follows the process illustrated in Figure A.1. Below is a detailed explanation to facilitate implementation. These streamlined steps focus on exploration, exploitation, and convergence to achieve the best solution.

- Initialization
  - Generate random solutions (countries). We use the solution representation of GA that was explained in Chapter 4.
  - Evaluate Fitness Using the Objective Function.
    - Each country, representing a potential solution, is modeled as a vector of decision variables within the search space.
    - The objective function evaluates how well a solution meets the problem's goal. For minimization problems, lower values of the objective function represent better fitness.
  - Select the Best Solutions as Imperialists.
    - Rank Solutions: Sort all countries based on their fitness. For a minimization problem, countries with the lowest fitness values are ranked highest.
    - Determine Imperialists: To determine the top n imperialist countries, the total number of countries C is ranked based on a predefined fitness criterion (e.g., cost). The top n countries with the best (lowest) fitness scores are selected as imperialists (leaders). Let Fitness(i) represent the fitness of country i, where $i \in \{1, 2, \ldots, C\}$.

      $$\text{Imperialists} = \{\text{Country}_{i_1}, \text{Country}_{i_2}, \ldots, \text{Country}_{i_n}\},$$

      where i_k corresponds to the k-th best fitness score, $k = 1, 2, \ldots, n$, (e.g., $n = 5$ for $C = 50$ countries).

    * Calculate Power of Imperialists: The power of each imperialist is inversely proportional to its fitness value. Use power values to proportionally assign colonies.

$$P_{\text{imperialist}} = \frac{1}{\text{Fitness Value}}.$$

  – Assign the Rest as Colonies to Form Empires.

    * Assign Colonies: Distribute the remaining countries as colonies among the imperialists. Each imperialist gets colonies proportional to its power relative to the total power. Randomly select colonies for each empire after determining the count.

$$N_{\text{colonies of imperialist}} = \frac{P_{\text{imperialist}}}{\sum P_{\text{all imperialists}}} \times \text{Total Colonies}.$$

    * Form Empires: Each empire consists of one imperialist (the leader) and assigned colonies (subordinate countries).
    * Balance Check: Ensure all colonies are assigned, and no country is left out.

- Colonial Movement: The process of improving colonies by moving them closer to the imperialist (better solution).

  – Process:

    * Determine Movement: Colonies take a step toward the imperialist, guided by the equation below. β is a random number between 0 and 1 that ensures variability in the step size. γ is a small random perturbation introduced to encourage exploration.

$$\text{New Position} = \text{Old Position} + \beta \times (\text{Imperialist Position} - \text{Old Position}) + \gamma,$$

    * Assimilation Movement: This ensures that colonies explore the space around the imperialist, reducing the likelihood of missing the global optimum.
    * Boundary Check: After movement, ensure the colony's position respects problem constraints (e.g., variable bounds).

  – Revolution: Introduce random changes to some colonies (analogous to a mutation in GA) to explore distant parts of the solution space. This ensures diversity and avoids stagnation.

- Assimilation and Takeover: Encourage improvement within empires by promoting better solutions. Process:

  - Evaluate Colonies: Compare each colony's fitness with the imperialist's. Fitness is measured using the objective function, where lower fitness indicates better performance (for minimization problems).
  - Colony Overtaking Imperialism: If a colony's fitness is better than the imperialist's fitness, the colony becomes the new imperialist. The previous imperialist becomes a colony within the empire. This ensures continuous improvement and prevents stagnation within an empire.

- Imperialistic Competition: Strengthen stronger empires by redistributing colonies and eliminating weak empires. Process:
  - Calculate Empire Power: The total cost of an empire is a combination of the imperialist's cost and a weighted sum of its colonies' costs. Empires with lower total costs are stronger.
  - Competition: Weak empires lose colonies to stronger empires. Redistribution prioritizes the best-performing colonies, which migrate to stronger empires.
  - Elimination: If an empire loses all its colonies, it is eliminated from the process. This focuses computational resources on promising solutions and simplifies the optimization process.

- Iteration: Repeat the steps above until the stopping criterion. In this thesis, the stopping criterion used is the maximum number of iterations.

- Output: Return the best imperialist's solution as the optimal or near-optimal solution.

Algorithm 1. Imperialist Competitive Algorithm

```
Initialize the countries and form the empires
while the stop condition is not satisfied do
    for all empires do
        Move the colonies toward the imperialist (assimilation)
        Make some colonies undergo a revolution
        if a colony is more powerful than the imperialist then
            The colony becomes the imperialist and vice versa (overthrow)
        end if
    end for
    if two empires are too close then
        Merge them (unification)
    end if
    Make imperialistic competition occur
    if there is an empire with no colonies then
        Eliminate this empire
    end if
end while
```

Figure A.1 Pseudo code of ICA

ICA Settings is explained in this paragraph. To effectively solve the optimization problem using the ICA, the algorithm's parameters were meticulously selected to balance exploration, exploitation, and computational efficiency. Three population sizes—50, 100, and 150 countries—were employed for simpler and more complex instances, respectively, ensuring adequate diversity and solution quality. The number of imperialists was set to 10% of the total population, resulting in 5 imperialists for a population of 50, 10 for 100, and 15 for 150 countries. This proportional allocation maintains a consistent level of competition among empires regardless of population size. In the context of the assimilation process in ICA, β and γ are crucial components influencing how colonies move toward the imperialist. These parameters ensure a balance between exploitation (refining the current best solutions) and exploration (searching for potentially better solutions in new areas). β is a random number between 0 and 1 that determines the proportion of the distance a colony moves toward the imperialist in a single step. γ is a small random value, drawn from uniform distribution $U(-\epsilon, \epsilon)$ (ϵ) is a very small value.

Additionally, a low revolution rate of 0.02 was applied to introduce controlled random changes, preventing the algorithm from becoming trapped in a local optima. The ICA was configured to run for a maximum of 100 iterations, striking a balance between achieving convergence and managing computational resources effectively. These settings, including the proportional number of imperialists, balanced assimilation and revolution rates, and appropriate population sizes, enabled the ICA to thoroughly explore the solution space, maintain diverse empires, and converge toward optimal or near-optimal solutions by simulating the

dynamics of imperialistic competition.