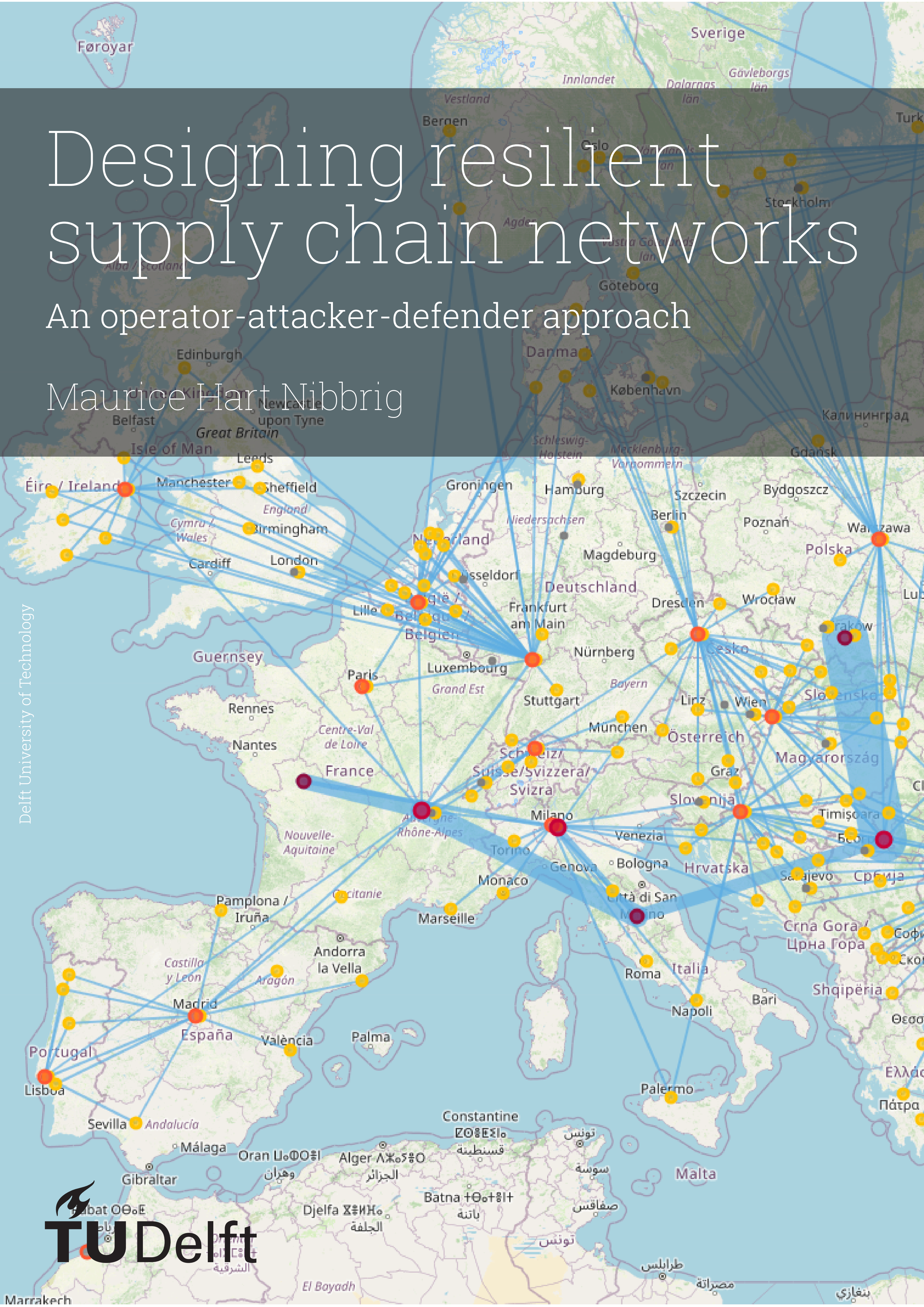
Designing resilient supply chain networks

An operator-attacker-defender approach

Maurice Hart Nibbrig



Delft University of Technology

Designing resilient supply chain networks

An operator-attacker-defender approach

by

Maurice Hart Nibbrig

Assessment:	M.Y. Maknoon (chair)
	S. Sharif Azadeh
	A. Bombelli
External supervisors:	J. Bleie
	M. de Rond
Project duration:	February - August, 2024
Faculty:	Faculty of Civil Engineering and Geosciences, Delft
	Faculty of Technology, Policy and Management, Delft

Cover: Map of the output of a run of our model on a random test case.
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

This thesis marks the completion of my MSc in Transport, Infrastructure, and Logistics at Delft University of Technology, and with it, my journey to becoming an engineer. My research on resilience in supply chain design covers a topic that's not only crucial across industries but also one that has deeply intrigued me. Looking back on the past six years at Delft, it's been a journey filled with challenges, breakthroughs, and connections with remarkable people who have become lifelong friends.

I want to express my sincere gratitude to the chair of my thesis assessment committee, Yousef Makoon, for our productive collaboration that shaped this work. I'm also thankful to committee members Shadi Sharif Azadeh and Alessandro Bombelli for their valuable feedback, which helped refine my research. A special thanks to Districon and Royal HaskoningDHV for giving me the chance to conduct my research within their organizations. I'm especially grateful to my external supervisors, Jesse Bleie and Martijn de Rond, for their professional insights, support, and for making my research experience much more enjoyable. My thanks also go to the entire Districon team for offering an insightful glimpse into the world of logistics and many memorable moments.

I'm ready for the next step.

Maurice Hart Nibbrig
Delft, September 2024

Summary

This study presents a comprehensive adaptation of the tri-level Operator-Attacker-Defender (OAD) model, tailored for the design of resilient supply chains capable of withstanding various disruptions, including supplier failures, production shortfalls, and transportation breakdowns. These disruptions pose significant challenges to a supply chain's ability to meet customer demand, making resilience a critical component of supply chain design. Originally introduced by Alderson et al. in 2011, the OAD model incorporates three levels of decision-making: the operator, who optimizes system performance; the attacker, who seeks to disrupt the system; and strategic design decisions that mitigate potential disruptions.

Our conceptual model involves a multi-commodity flow network, which includes a defined performance metric to evaluate the supply chain's performance. To resolve the complex tri-level model, we employed a decomposition-based solution strategy, breaking down the problem into master problem and an attacker sub-problem. Extensive computational experiments have been conducted to test the model's tractability across various supply chain scenarios, demonstrating its ability to handle complex, real-world situations effectively.

In addition to the computational experiments, we conducted a case study on a global pharmaceutical supply chain, focusing on its vulnerability to climate-related disruptions. This case study provided a practical demonstration of the OAD model's effectiveness, showing how it can identify weak points in the supply chain and suggest optimal strategies to enhance resilience. The model's ability to balance trade-offs between operational efficiency, costs, and resilience makes it a valuable tool for supply chain managers. From a computational standpoint, the model's performance is influenced by several factors, including the size of the decision network and the complexity of the value chain. Larger decision networks, which involve more system design variables, typically lead to longer solve times due to the increased complexity. However, more complex value chains do not always result in longer solve times; instead, they may be easier to disrupt, enabling quicker identification of optimal solutions. Additionally, the interplay between attack and defense budgets significantly impacts solve time, with larger budgets generally expanding the solution space and increasing computational effort. The comparison between deterministic optimization and simulation approaches reveals the deterministic method's superior efficiency in identifying optimal disruptions. The deterministic attacker sub-problem outperforms the simulation-based approach, particularly as the attack budget increases, underscoring its value for critical supply chain analysis.

Overall, our findings confirm that the OAD model is a robust and computationally tractable framework for studying resilience in supply chains. It meets the objectives set out at the beginning of the study, including the development of a valid OAD model, the exploration of various supply chain topologies and complexities, and the analysis of different disruption types and operational responses. Extending the model to include a temporal dimension could provide deeper insights into how time-related factors like delays and expiration dates impact supply chain resilience. Incorporating such elements would allow the model to support not only strategic-level decisions but also more tactical-level considerations. This would be interesting further research. Finally, the test case study highlights the importance of accurately defining attack costs and budgets for identifying vulnerable parts of the supply chain. While our study suggests using climate hazard data as a basis for determining these costs, further research is needed to develop more robust methods grounded in theoretical risk analysis. Such advancements would enhance the model's ability to inform strategic design decisions and contribute to the ongoing development of resilient supply chains.

Contents

List of Tables

1 Introduction

The global chip shortage of the early 2020s, intensified by geopolitical tensions and the COVID-19 pandemic, exposed significant vulnerabilities in global supply chains (SCs) and led to widespread production delays across various industries sweney2021chipshortage. Whether arising from geopolitical conflicts or natural occurrences, such as the early 2020s pandemic, recent disruptive events highlight the vulnerability of supply chains and their far-reaching effects on global trade, manufacturing, and the economy. In the complex and dynamic world of global commerce, supply chains play a pivotal role in ensuring the smooth flow of goods and services. The growing interconnectedness and interdependence of SC networks expose them to uncertainties and disruptions, necessitating a more robust integration of resilience into their design and operation. The increasing effects of climate change further underscore the importance of researching resilient supply chain strategies to navigate uncertainties and effectively mitigate the impact of disruptions.

This document presents a research proposal focused on investigating the integration of resilience into the design and operation of SC networks using a mathematical programming-based solution. The primary goal is to formulate a three-level Mixed-Integer Programming (MIP) model, termed the Operator-Attacker-Defender (OAD) model. This model integrates base operator, attacker, and defender components into a single optimization framework, tailored to simulate the impact of potential disruptions and incorporate mitigation strategies at strategical and operational decision levels. A more detailed exploration of the three progressively expanding models constituting the OAD model will follow in subsequent sections of this proposal. The overarching aim of this research is to contribute valuable insights to the field of resilient SC optimization, assisting organizations in developing SCs that can effectively manage and respond to the most critical disruptions, thereby enhancing their overall resilience. The central research question guiding this study is: *How can resilience effectively be integrated into the strategic design and operation of supply chain networks in the face of disruptions?*

1.1. Problem significance

The significance of the problem this study aims to address is motivated for multiple reasons. The attacks on the Suez Canal highlight the geopolitical dimensions that can severely impact critical trade routes. Such incidents underscore the need to not only address traditional disruptions like natural disasters but also to develop strategies against intentional threats by intelligent agents, which have become more prevalent in the contemporary global landscape, showing an uptick in geopolitical tensions. Furthermore, the cascading effects of disruptions extend beyond immediate economic losses. They can lead to long-term repercussions such as erosion of customer trust, damage to brand reputation, and geopolitical ramifications (Manners-Bell, 2014). The interconnectedness of supply chains means that a disruption in one part of the world can have a ripple effect, affecting businesses and consumers across various regions. Therefore, understanding and mitigating the multifaceted consequences of disruptions are crucial for maintaining the stability and integrity of global supply chains. In the context of the ongoing global shift towards sustainability, there is an increasing focus on the environmental impact of supply chain disruptions. Climate change-induced events, such as extreme weather events and rising sea levels, pose a significant threat to the resilience of supply chains. Incorporating environmental considerations into the design and operation of supply chain networks is not only a matter of risk management but also aligns with the broader societal goals of sustainability and environmental stewardship.

Resilience in supply chain system design is explored extensively in the literature with various approaches including scenario-based stochastic programming, robust optimization, and other non-deterministic models to address uncertainties and disruptions. The literature review has revealed some limitations in the current state-of-the-art. Existing models often struggle with representative scenario generation, conservatism-tractability trade-offs, and insufficient consideration of both natural and deliberate disruptions (Alderson et al., 2011). Many models focus on specific aspects: either on mitigation or post-disruption decisions, without offering an integrated decision-making framework (Govindan et al., 2017). Trade-offs between cost, operational performance, and resilience are often insufficiently understood through the proposed approaches. The OAD model introduced above aims to overcome these challenges by incorporating game theory principles, forming a comprehensive model that integrates offensive and defensive strategies. It provides deterministic insights into critical network components and allows for a nuanced analysis of operational and defensive investment trade-offs. Adapting this model to the challenges of resilient supply chain design and operation is proposed as a valuable avenue for advancing the field.

1.2. Document structure

The structure of this research proposal unfolds as follows: after this introduction, a literature review in section 2 will delve into existing research on disruptions and resilience in supply chains. Following that, section 3 will outline the proposed research goals and contributions, while section 4 will introduce the conceptual and mathematical models, providing an overview of the three-level optimization model. Section 5 will then present a resolution strategy. Finally, the model results will be presented through computational experiments in 6.1 and a realistic case study regarding climate resilience in a global pharmaceutical company in 6.2.

2
Literature review

This section of the research proposal consists of a literature review on resilience in supply chain network design (SCND) and operations. Following a brief introduction to various SC principles, we explore supply chain resilience. We, examining sources of uncertainties in SCND, as well as disruptions and risk management. The subsequent segment entails a review of the literature on optimisation of resilient supply chains under conditions of uncertainty. Finally, we investigate the applications of the attacker-defender model, a optimisation model for network defence, within various contexts.

A supply chain is a, usually complex, logistics network comprising facilities that transform raw materials into finished products and subsequently distribute them to end consumers or customers (Harrison & Godsell, 2003). Concurrently, supply chain management (SCM) focuses on optimizing the flow of goods within the supply chain for maximum efficiency (Wieland & Wallenburg, 2011). In SC management, decisions involve strategic choices like network design and tactical and operational considerations like production scheduling, location allocation and product flow. Used products can rejoin a supply chain to be recycled in reverse supply chains (RSC). Although interesting, the rest of this literature review will focus on forward SC. Supply chain network design (SCND) is an important planning problem in SCM involves optimizing the configuration of production facilities, distribution centers, and transportation routes to enhance overall efficiency and meet strategic objectives. SCND can be likened to a network flow problem where nodes represent supply chain entities, such as suppliers, productions sites, distribution centers, and edges depict transportation paths (Zhen et al., 2016). The optimization goal of such a problem is generally to efficiently allocate resources, minimize costs, and meet logistical demands within the interconnected network.

2.1. Resilience in supply chains

Wieland et al. (2021) define supply chain resilience as ”the capacity of a supply chain to persist, adapt, or transform in the face of change” (Wieland & Durach, 2021). Resilience involves the ability to navigate through various changes. It is more about the inherent characteristics and robustness of the system, rather than the external risks (Sheffi, 2007). The older perspective on supply chain resilience, engineering resilience, treats the supply chain as a controllable system with a focus on quick restoration to the initial setup after disruptions. In contrast, a newer perspective, socio-ecological resilience, views the supply chain as a dynamic, fluid entity capable of continuous adaptation and even transformation in response to external conditions. The system need not return to its initial stage (Folke, 2006).

Uncertainty in SC network design

The supply chain network design (SCND) problem involves multiple decision-making levels. At each level, there may be uncertainties during the design process. SCND problems encompass parameters like costs, demand, and supply with inherent uncertainty. Additionally, major disruptions such as natural disasters or economic crises can impact SC networks. The aim is to configure SCND to excel under these uncertain scenarios, according to the decision-maker performance criteria. Rosenhead et al., 1972 categorize uncertain SCND decision-environments into three groups: (1) environments with known probability distributions for random parameters, often modeled using a scenario approach, (2) environments with random parameters but without information on their probability distributions, often modeled using robust optimization models, and (3) fuzzy decision-making environments involving ambiguity and vagueness in objective function components, variable coefficients, and constraint satisfaction (Rosenhead et al., 1972). Examples of SCND models in the first group include E. Huang and Goetschalckx, 2014 and Kumar and Tiwari, 2013, Aliakbar Hasani and Nikbakhsh, 2015 and Zokaee et al., 2017 in the second group, and Mousazadeh et al., 2015 in the third group.

SC networks involve various facilities organized into layers or echelons and material flows from suppliers to customers. Layers or echelons consist of facilities with similar tasks. Studies differ on the number of location layers over which decisions are made, e.g., Subulan et al., 2015 use a single layer, Yılmaz Balaman and Selim, 2014 use two, and Babazadeh et al., 2017 three. Studies also differ on the number of different product types: single (Q. Li & Hu, 2014) or multiple commodities (Subulan et al., 2015). Different material flow configurations may be considered. Some papers consider only single-sourcing, but Zeballos et al., 2014 considers flows within a same layer, and Tong et al., 2014 also includes direct product flows between customers and suppliers. Real-life applications often require addressing multi-product problems. The SCND may involve a different number of decision periods: while most models consider only a single period (e.g., E. Huang and Goetschalckx, 2014, Q. Li and Hu, 2014), some studies explore multiple tactical/operational (e.g., Pasandideh et al., 2015, Mousazadeh et al., 2015) or strategic time periods (Pimentel et al., 2013). SCND decisions may consider different time spans: strategic (multiple years), tactical (multiple months), or operational decisions (hourly to weekly). Parameters that may be subject to uncertainty in SCND include the product demand, various costs (transportation, production, operations), facility and transport link capacities or supplier or producer capacities. Uncertain parameters may also include the parameters of distribution functions (e.g., of demand), processing times (production, transportation, supply) or uncertainties in the actual nature of the performance goal (Pasandideh et al., 2015). The up-time (availability) of network links and facilities are also subject to uncertainty due to risks of disruptions (Govindan et al., 2017).

Disruptions and risk management in SC

There is a lack of clear consensus regarding the meaning of the concept of supply chain risk. Risk is usually defined around the potential loss in a supply chain's objectives due to uncertain variations triggered by events (Ho et al., 2015). In the previous section 2.1, sources of so-called operational risks have been listed. Operational risks are risks that stem from intrinsic uncertainties in supply chain elements. A second type of risk category found in the literature on the SCND problem is disruption risks. Disruption risks result from events such as natural or human-caused disasters that have undesired effects on the supply chain's goals and performance. Disruptions can affect the functionality of supply chain elements, either partially or completely, for an uncertain duration (Tomlin & Wang, 2011).

According to Fattahi et al., 2020, SC resilience to disruptions involves designing networks "that can return to their initial or a more desirable state" after disruptions (Fattahi et al., 2020). If a SC network can operate efficiently in normal conditions as well as during disruptive events, it can be considered resilient according to most of the literature. It is difficult to measure resilience; decision-makers ought to define their indicator according to their needs and views (Christopher & Peck, 2004). Indeed, there are different views, or paradigms, regarding risk management in SC. For some organizations, a resilient SC is a responsive one: it prioritizes flexibility and adaptability to events (Harrison & Godsell, 2003). For others, it involves developing green SCs, which incorporate environmental considerations (such as Babazadeh et al., 2017, and sustainable SCs, which balance multiple aspects - economic, environmental, and social - to design SCs that also take into account the needs of future generations (**govindan2022supplyREVIEW**). Resilience principles also apply to contexts outside of businesses, such as disaster relief (Liu & Guo, 2014). Such SCs are described as humanitarian. Snyder et al., 2014 have created a comprehensive overview of SC disruptions and various mitigation strategies. Strategies against disruptions can involve preventive actions (also called mitigation), or responsive actions (or contingency) during and directly following a disruption. Some mitigation strategies against disruptions, according to their overview, include facility fortification, involving the selection and fortification of specific facilities within the SC network, strategic stock management, wherein inventory is strategically held across different SC layers to meet customer demands and support manufacturing processes, and sourcing strategies (multiple and backup sourcing) where multiple suppliers are used simultaneously or as a backup during disruptions (Snyder et al., 2014).

2.2. Resilient supply chain optimisation

Various approaches to optimising the design and operations of supply chains when faced with uncertainties, such as disruptions, can be found in the literature. These different approaches often reflect the decision-environment groups distinction introduced in section 2.1.

Approaches to SC optimisation under uncertainty

A first approach to optimise SCND with uncertain parameters is by using scenarios in a scenario-based stochastic program. Scenarios need to be generated in a set, and they have known probabilities. In two-stage stochastic programs, there are separate decision-makings before and after the realization of random parameters. In the first stage, strategic decisions are made. The second decision stage involves optimising operational decisions based on an expected objective. In multi-stage stochastic programs, multiple decision stages are included, with a sequence of random parameters (Pimentel et al., 2013). In this approach, one obtains a scenario-tree, which branches at each decision stage (Fattahi et al., 2018). However, generating accurate and representative scenarios poses a challenge, potentially leading to large-scale optimization problems as the scenario tree can explode in the numbers of scenario combinations (Kazemi Zanjani et al., 2016).

Another approach to SCND, where the model parameters have a stochastic continuous known distribution, involves solving a mixed-integer nonlinear programming (MINLP) model. The non-linearity present in most of these models requires specific solving methods that improve solve efficiency. This model type is usually tackled using Lagrangian Relaxation (LR) embedded into the Branch and Bound (B&B) algorithm, such as done in Tanonkou et al., 2008 or Yongheng et al., 2014. Occasionally, a column generation algorithm is included, such as in Shen et al., 2003. In cases where certain constraints in SCND optimization problems must be satisfied with a specified probability or reliability level, the chance-constrained programming approach is employed. This technique involves modeling constraints to meet pre-specified probabilities (S. Huang et al., 2023)), particularly suitable for scenarios where the availability or reliability of facilities or transportation links is considered with predetermined probabilities. Traditional optimization approaches often optimize the expected value of an objective (Klibi et al., 2010)). However, when there are large variations in stochastic parameters, a risk measure may be defined, see Hamed Soleimani and Kannan, 2014. This is a function that maps a random outcome to a real value and is often used in the context of economics.

Robust optimization is another way of handling uncertainties in SCND in the literature. It involves finding solutions that are robust given an area of uncertainty for certain parameters. Robust SCND models may use discrete scenarios, others use a certain interval uncertainty. The idea is usually to optimize for the most optimal solution in the worst scenario: min-max cost/regret objectives are used (see Mozafari and Zabihi, 2020 or Zokaee et al., 2017), and various risk measures are applied to ensure robustness. However, achieving a balance between conservatism in robust solutions and maintaining tractability poses a challenge (Keyvanshokooh et al., 2016). Fuzzy mathematical programming is used where uncertainties are characterized by vague goals, soft constraints, and ambiguous coefficients (Soleimani et al., 2017). Utilizing flexible programming, this approach is particularly beneficial when decision-makers have fuzzy preferences or lack precise information.

SC optimisation against disruptions

Disruptions can essentially be viewed as a specific type of yield uncertainty. Yield uncertainty represents uncertainty in the delivered product quantity, often modeled as a stochastic variable influenced by customer order quantity. Similarly, capacity uncertainty can be depicted as a stochastic variable, generally independent of order quantity. Notably, disruptions differ from yield and capacity uncertainties as they are typically discrete events, while the latter variables are treated as continuous (Snyder et al., 2014). Most models distinguish between up/wet and down/dry SC network states for normal and disrupted supply situations respectively, assuming that a network can transition between both states in time-periods following an exponential distribution (Ross et al., 2008). Redundancy is key in managing supply-side disruptions, and various optimization approaches exist in the literature.

The first strategy involves optimizing inventory replenishment, determining the optimal timing, quantity, and sourcing of material orders within the network. While models for normal inventory management exist, those addressing disruptions usually involve added complexity. These models can make periodic (Karakatsoulis & Skouri, 2023) or continuous (Qi, 2013) inventory decisions. They are often based on the Economic Order Quantity (EOQ) model from economics, with integration of a disruption component, forming an Economic Order Quantity with Disruptions (EOQD) model (Karakatsoulis et al., 2024).

Another approach focuses on modeling flexibility in sourcing. When dealing with unreliable suppliers susceptible to disruptions, retailers may place similar orders with multiple suppliers, a practice known as routine sourcing. Hosseini et al., 2019 proposes a model for optimizing flexible supplier selection. Alternatively, if backup suppliers are only used when a disruption occurs, this is referred to as contingent rerouting and is explored in Birge et al., 2022. Models considering backup sourcing must also include decisions regarding corresponding backup capacity requirements. Alternatively, Gupta et al., 2021 suggests a demand-side model optimizing pricing adjustments during capacity disruptions.

Additionally, facility location models can also address disruptive events. Location decisions may involve rerouting post-disruptions (a contingent strategy), or more strategic decisions that have to be made in advance. Models allowing reallocation of customers to facilities during disruptions, such as Wang et al., 2023, may require optimization of facility locations beforehand. Other strategic decisions involve fortification, determining which location should receive improvement investments before disruptions (X. Li et al., 2018). Some models focus on external parties, optimizing insurance strategies and fostering cooperation and competition between suppliers (Chakraborty et al., 2019).

Attacker-defender model

An intriguing possible approach to resilient network design is the attacker-defender model, also known as the operator-attacker-defender (OAD) or defender-attacker-defender (DAD) model. This approach is found in various other domains where a certain network has to be designed and protected against harm. This game theory-based approach introduces multiple decision layers to the optimization model, considering both offensive actions, i.e., disruptive events that harm normal operational performance, and defensive strategies integrated under a unified optimization objective. The attacker-defender model has found applications in diverse domains where network-like systems require protection against external disruptions.

The paper by Alderson et al., 2011 presents a defender-attacker-defender (DAD) model tailored for planning defenses in infrastructure systems, specifically enhancing resilience against intelligent adversary attacks on transportation networks. The DAD model involves the defender selecting infrastructure investments, the attacker observing and executing an attack, and the operator (defender) assessing both, managing the system to minimize operating costs. The authors develop a decomposition algorithm for solving instances of the DAD model and showcase its application on a transportation network example, demonstrating the versatility and applicability of the approach (Alderson et al., 2011).

Addressing the vulnerability of power grids to terrorist attacks, Yuan et al., 2014 propose an enhanced solution framework based on the defender-attacker-defender model. They introduce a Column-and-Constraint Generation algorithm to efficiently solve their DAD model, outperforming existing methods. The study emphasizes the superiority of the DAD model in improving grid survivability over a traditional attacker-defender model, providing valuable insights and an efficient algorithm for enhancing protection in critical infrastructure networks (Yuan et al., 2014). Again in the power system protection domain, Xiang and Wang, 2019 present the Multiple- Attack-Scenario Defender-Attacker-Defender (MAS DAD) model to address uncertainties in defending electric power systems. This extended model focuses on optimizing defensive resource allocation to minimize damage in the face of uncertain attacker capabilities. The use of a Column-and-Constraint Generation algorithm allows for the decomposition of the MAS DAD model, providing an interesting solution method when dealing with uncertainties in the attacker budget (Xiang & Wang, 2019).

Additionally, Xu et al., 2016 integrates a defender-attacker game with military supply chain risk management, revealing the benefits of risk management tools for the defender. However, certain protection strategies were found to have no impact on the attacker's resource allocation. This emphasizes the complex dynamics involved in defender-attacker interactions within supply chain systems (Xu et al., 2016).

2.3. Research gap and contribution

The literature reviewed indicates that resilience in the design and operation of supply chain systems is a well-studied topic with various approaches employed. While enlightening, the approaches introduced in 2.2 and 2.2 have limitations identified through our analysis and by respective authors. Scenario-based stochastic programming faces challenges in generating representative scenarios, leading to large-scale optimization problems. The challenge of representative scenario generation and its impact on optimization efficiency and model accuracy is a drawback (Fattahi et al., 2018). Robust optimization struggles to balance conservatism and tractability. Non-deterministic models depend on hard-to-find disruption probability data, diminishing their validity. Many models focus solely on mitigation or post-disruption decisions. Resilience is found at the interplay of both; an integrated decision model would be beneficial (Govindan et al., 2017). Furthermore, Insights into cost, operational performance, and resilience trade-offs are often limited. A truly resilient system, as argued by Alderson et al., 2011, must perform well for all events, not just the most likely ones. Probabilistic models optimising on expected objectives have difficulty accounting for this. Parts identified as ”most critical” by stochastic models differ from worst-case analysis. Finally, they argue that both natural and deliberate disruptions should be involved and that sensibly combining both disruption types remains an unexplored research area (Alderson et al., 2011).

The Attacker-Defender model from 2.2 offers a promising solution. Applied successfully in various domains, its application to resilient supply chains and the SCND problem is an unexplored research gap. This game theory-based model considers both offensive actions (disruptions) and defensive strategies, allowing for in-depth analysis of operational and defensive investment trade-offs. Unlike probabilistic approaches, the attacker component provides deterministic insights into critical network components without relying on unreliable probability data. The model’s setup permits flexible expansion by incorporating and comparing evermore disruptive impacts and defensive strategies. All in all, adapting the attacker-defender model to address the unique challenges and dynamics of resilient supply chain design and operation opens a valuable avenue for research, promising to advance the field of supply chain resilience, and shaping the remainder of this proposal.

3 Problem statement

The primary aim of this research is to develop an integrated mathematical framework for enhancing resilience in supply chain network design and operation in the face of disruptions. By applying a quantitative approach, we aim to tackle the Supply Chain Network Design and Operation Problem (SCDOP) using the Operator-Attacker-Defender (OAD) model. Our version of the model will be a three-level Mixed-Integer Programming (MIP) framework that incorporates supply chain system design decisions, potential disruptions, and mitigation strategies into a unified optimization framework.

3.1. Research question and objectives

The main question guiding this research is: *How can resilience effectively be integrated into the strategic design and operation of supply chain networks in the face of disruptions?*

This will be approached using the following objectives:

Obj 1 Develop a valid and computationally tractable OAD model adapted for studying resilience in supply chains.
Obj 2 Be able to explore a multitude of supply chain topologies and value chain complexities.
Obj 3 Be able to explore a multitude of disruption types and operational responses.

The study will focus on forward supply chains, as opposed to reverse logistics/recycling supply chains, featuring a unidirectional flow of products from suppliers to customers through a network of locations. This model concentrates on the strategic level (decisions regarding system design and aggregate network flows) and does not involve a temporal dimension. However, for potential future research, section 7.2 will introduce an adaptation that incorporates a temporal component, allowing for more tactical-level studies.

3.2. Interdiction and OAD models

In an interdiction model, two opposing agents, the operator and the interdictor or attacker, engage in a strategic game where the operator aims to maximize performance while the attacker seeks to minimize it. This sequential decision-making process mirrors the Stackelberg game, where a leader makes decisions first, considering the follower's potential responses. The interdictor identifies strategic targets to disrupt the operator's activities, aiming to minimize their effectiveness. OAD models extend this concept by introducing a third level where the operator allocates defensive resources to accomplish its objectives while mitigating the impact of potential attacks. In our supply chain model, this is represented through three decision layers within a single mathematical program:

1. The **base operator layer**, in which the *operator* determines the optimal system operation and performance, given the system design and operational setting,
2. The **attacker layer**, in which the *attacker* determines the worst-case operational set (set of events) that minimize optimal performance,

3. The **design/defender layer**, in which the *operator* chooses the best system design, in anticipation of a worst-case operational setting.

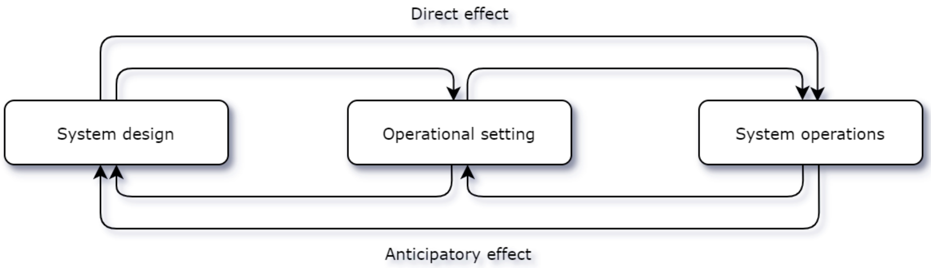


Figure 3.1: Tri-levels of interdiction interaction.

$$
\begin{aligned}
&[\textbf{Generic OAD}] && && && \qquad (3.1)\\
&\underset{\boldsymbol{x_D}}{min}\ \underset{\boldsymbol{x_A}}{max}\ \underset{\boldsymbol{x_O}}{min} && \Gamma^{performance}(\boldsymbol{x_O}, \boldsymbol{x_A}, \boldsymbol{x_D}) \\
&\text{s.t.} && \boldsymbol{x_O} \in X_O\ (\boldsymbol{x_A}, \boldsymbol{x_D}) \\
& && \Gamma^{attacks}\ (\boldsymbol{x_A}) \leq bdg^{att} \\
& && \Gamma^{design}\ (\boldsymbol{x_D}) \leq bdg^{def}
\end{aligned}
$$

We can formulate this in a generic mathematical program format like in Eq. 3.1. The operations $\boldsymbol{x_O}$, operational setting $\boldsymbol{x_A}$ and system design $\boldsymbol{x_D}$ decision variables compete in a tri-level objective function to minimize or maximize a performance cost function $\Gamma^{performance}$. $\boldsymbol{x_O}$ is constrained to the set of valid system operations, a function of $\boldsymbol{x_A}$ and $\boldsymbol{x_D}$. The interdictor's cost of attacking $\Gamma^{attacks}$ is constrained to an attack budget bdg^{att}. The design costs Γ^{design} are constrained by a design budget bdg^{def}. Achieving **Obj 1** will involve formulating a version of the model suited for supply chains, that remains computationally tractable.

3.3. Value chains and supply chain topologies

A supply chain is a network of facilities enabling the flow of commodities, handling their production, transformation, and distribution. These systems vary in size and complexity. **Obj 2** requires that the proposed model adapts to various network configurations. The physical, spatial layout of system locations and their interconnections is referred to as the supply chain's topology. We identify four location types: suppliers, producers, warehouses, and customers. Conceptually, at the strategic level, they only differ in how they are involved in commodity flows: suppliers are sources of goods, customers are sinks thereof, producers transform goods and warehouses handle transient flows. The network topology varies in possible connections, including intra-layer and direct customer connections. Transportation connections are often multi-modal, depending on the connected layer types.

More fundamental than the supply chain's topology is the underlying value chain. The value chain encompasses the activities a business performs to deliver goods to end customers. Deliverables may result from a single production step with few raw materials or a complex network of interconnecting steps involving many suppliers and production sites. In complex value chains, goods can follow multiple paths to the end customer. Figure 3.2 depicts a complex value chain in the iron/steel goods industry. Complex chains have more points of failure, making resilience studies more pertinent. Our model should remain agnostic to the value chain layout, as dictated by **Obj 2**.

3.4. Cascading disruptions and operational responses

Supply chain resilience is "the capacity of a supply chain to persist, adapt, or transform in the face of change" (Wieland & Durach, 2021). Disruptions are a form of supply chain uncertainty, typically through

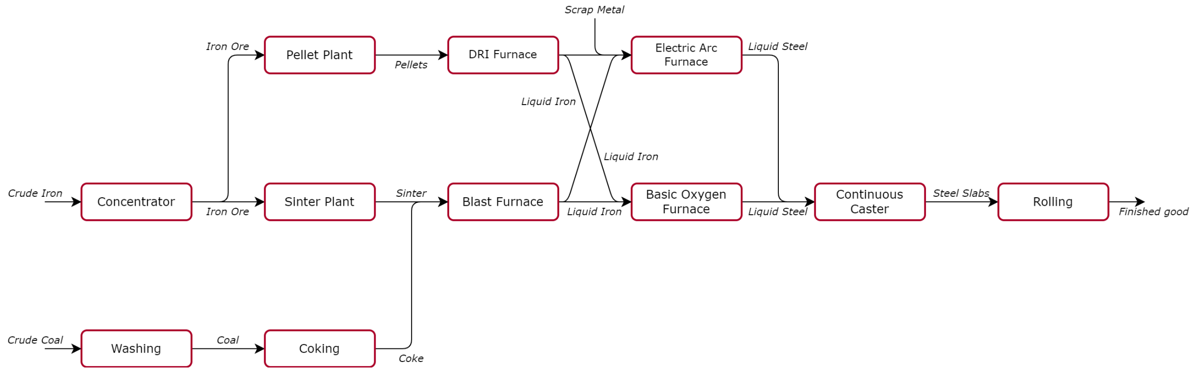


Figure 3.2: The value chain of the iron/steel goods industry.

discrete events. Disruptions affect yield uncertainty, the uncertainty in the quantity of delivered product. Regarding **Obj 3**, this paper focuses on supply-side disruptions, including partial or full reductions in facility and link capacities, but not demand-side uncertainties like customer demand fluctuations. Disruptions can disable facilities (e.g., due to war) or processes at suppliers (supply failure), producers (machine breakdown), or warehouses. Transport links can also be disabled (e.g., Suez Canal closure). Strategies against disruptions include preventive actions (mitigation) and responsive actions (contingency) after a disruption. Mitigation strategies modeled include flexible/multi-sourcing for unreliable suppliers and redundant production/storage capacity planning. Post-disruption rerouting of commodity flows is a contingent strategy, the success of which also relies on prior strategic decisions. Our model should enable the study of cascading effects in complex value chains (see 3.3).

4
Model

This study introduces a mathematical programming-based solution to integrate resilience into supply chain network design and operation. The proposed framework, inspired by the work of Alderson et al., 2011, is a three-level Mixed-Integer Programming (MIP) model termed the Operator-Attacker-Defender (mOAD) model. It comprises three hierarchical decision levels: the operator, the attacker, and the defender. In section 4.1, we introduce a conceptual model for the supply chain, which forms the foundation for our approach. Subsequently, in 4.2, we construct the tri-level mathematical program step-by-step. The resolution approach for this complex model is detailed in the subsequent chapter, section 5.

4.1. Supply chain conceptual model

4.1.1. Components of a production-distribution supply chain

Conceptually, a supply chain consists of various components that interact with each other. We define the following components. **Commodities** are the goods traversing the network, encompassing raw materials, intermediates, and finished products. This system does not differentiate between these categories, viewing them all as items flowing through the network, subject to transformation and distribution. **BillOfMaterials** describe a production step, similar to a recipe, detailing how one or more input commodities can be transformed into one or more output commodities. They include data on the amounts of each input and output commodity the production step consumes and produces. Each commodity has its unit of measurement (e.g., kg, or $each$). The supply chain system consists of various **Locations**, which include the following four types: **Suppliers**, **Producers**, **Warehouses**, and **Customers**. Suppliers provide commodities to producers, who transform them according to various BillOfMaterials. Warehouses store commodities before distribution to customers. Customers exhibit demand for one or more types of commodities and are served from warehouse locations. Producers may be interconnected to form more complex chains of production steps.

Locations are connected to each other via **Links**. A single link may connect a pair of locations and has a distance based on the coordinates of the locations it connects. A link may have a flow of one or more commodities, with the flow volume measured in the commodity's unit. One or more **TransportModes** may be available on a link. TransportModes can be defined individually per link, depending on the input data. For simplification, links connecting similar location types usually have the same available modes. The number of trips performed per TransportMode on a link are referred to as link loads. A **Process** connects a location to a commodity or BillOfMaterial and is also input data. A **Supply** process defines the supply of a commodity from a supplier. Similarly, a **Production** process defines the production of a BillOfMaterial process at a producer, and a **Storage** process defines the storage of a commodity at a warehouse. The process definitions include the associated process capacity and unit cost. A TransportMode trip on a specific link has an associated trip cost. In our model, we assume that a trip cost between two locations using a particular mode can be calculated as the sum of a fixed mode-dependent trip-start cost and a variable distance-based cost. For simplicity, it is assumed that on a link, different commodities may share a trip using the same mode. Furthermore, each TransportMode has

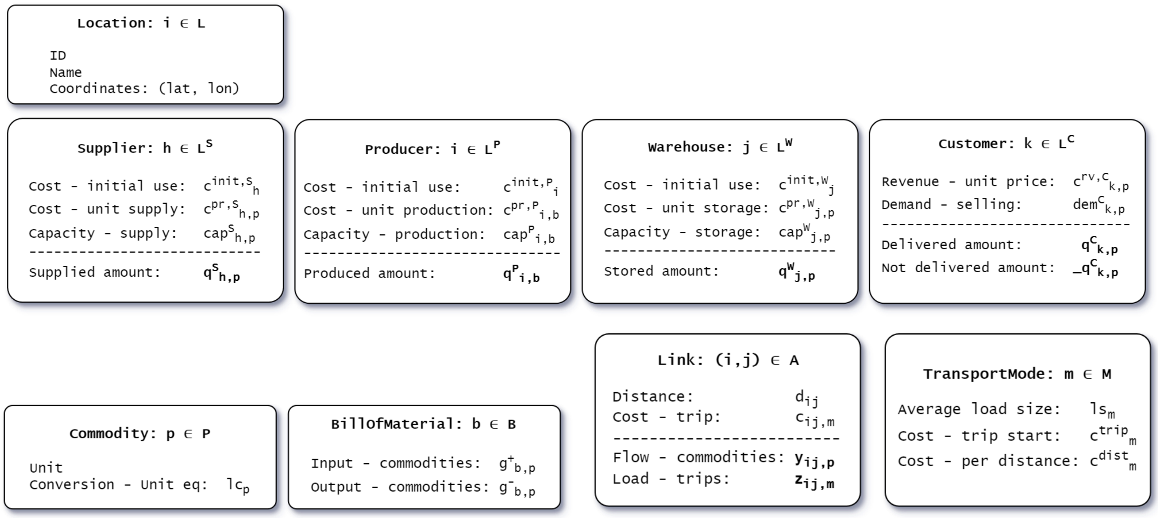


Figure 4.1: Overview of the conceptual model components and their attributes.

an average load size, defined in a standard unit. A commodity's unit can be converted to this standard unit using the commodity's unit load conversion attribute.

We define the network consisting of one or more suppliers and one or more interconnected producers as the **Production Chain**. This network represents (of part of) the value chain - the progression of activities used to deliver goods to an end customer. The Production Chain is defined up to the warehouse. A warehouse could technically be connected to a supplier directly. The network consisting of warehouses and customers is referred to as the **Distribution Chain**. Connected together, they form the full supply chain, a flow network where resources flow between nodes. This chain facilitates the movement of goods from storage to end-users, ensuring the fulfillment of demand.

4.1.2. Supply Chain Decision Network

The model will make decisions on the **Supply Chain Decision Network** G^{SC}, a non-complete directed graph of all possible locations and links. The Supply Chain Decision Network is constructed by combining the input data on the physical network, i.e. the locations and transport links, and the Production Graph G^{PROD}. The Production Graph is a representation of the value chain of a supply chain. It is an abstract graph where the nodes represent specific production steps and the arcs represent commodity flows between those steps. Figure B.1 depicts an example of a Production Graph.

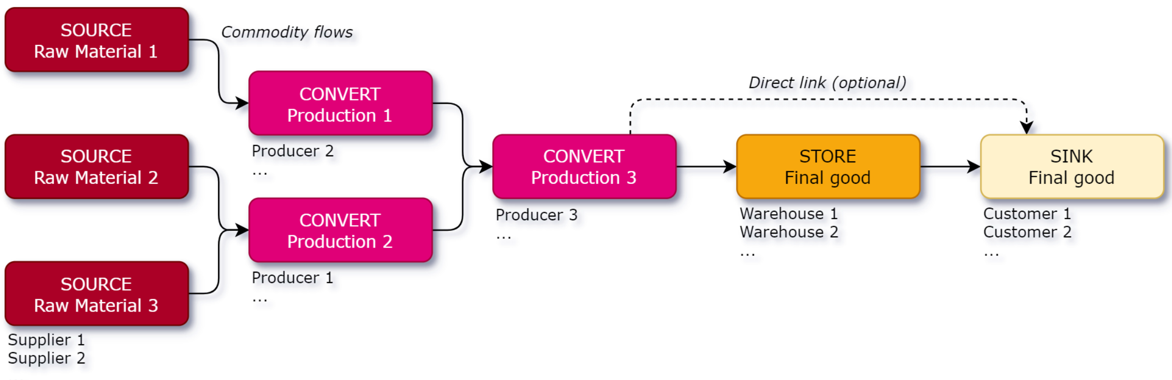


Figure 4.2: Example Production Graph G^{PROD} (with some locations listed).

Figure B.3 depicts a Supply Chain Decision Network and the model's parameters. A procedure for constructing G^{PROD} and G^{SC} can be found in Appendix B.

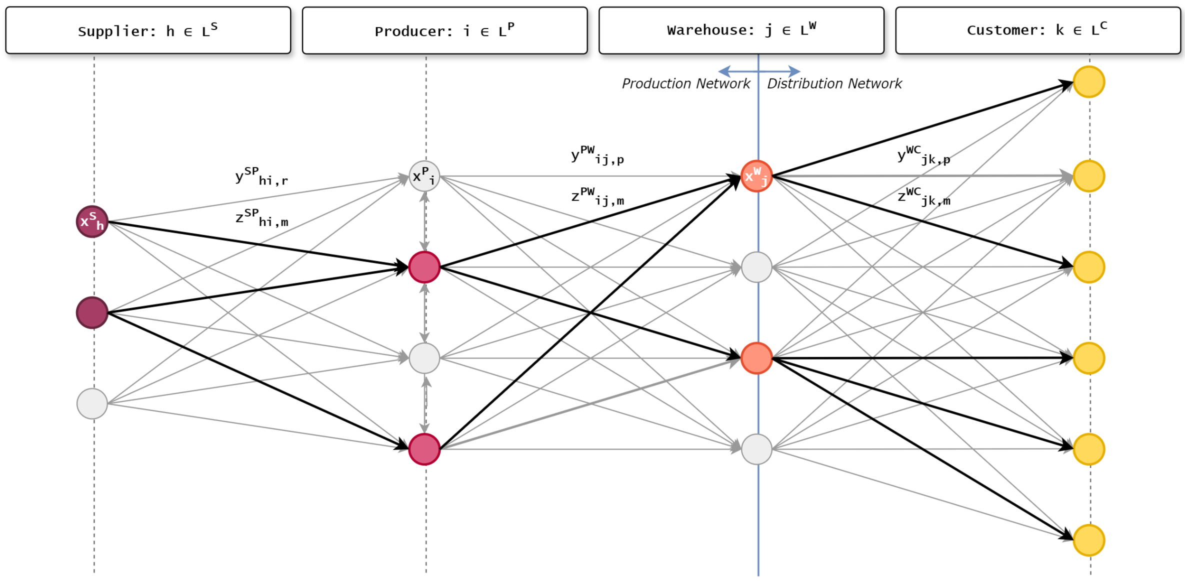


Figure 4.3: Overview of the Supply Chain Decision Graph $G^{SC}(L, A)$, our supply chain model.

4.1.3. System operational performance

To study the resilience of the supply chain system, its operational performance needs to be defined. Operational performance measures how well the system functions. In the context of supply chain resilience, it will be used to assess the impact of various disruptive settings compared to the normal operational setting. The operational performance will be a function of various operational costs incurred over a defined lifetime. These costs include supply, production, storage, transportation, and fixed costs. However, this is not sufficient to measure resilience. A disruption affects the capacity of a supply chain to deliver goods to customers, for example, due to reduced capacity at a location or transport link. Therefore, the operational performance will also include a penalty cost for the failure to deliver demand to customers.

4.2. Mathematical model construction

Building on the conceptual model introduced in the previous section, we now construct the tri-level optimization model. This model will be built-up step by step, starting by introducing the model parameters, decision variables, objective and constraints relating the operations of a supply chain. Following that, the model will be expanded by including the variables and constraints relating to the attacker component. The model is then similarly expanded further to include the design / defence components, in order to finally form the complete tri-level model.

4.2.1. Base operator model

The operator components should model the decisions an operator would make in order to efficiently operate a supply chain, assuming a given system design and operational setting. These decisions involve managing the flow of commodities across the Production / Distribution networks, including the various production processes at the different facilities and the transportation between them. As will be made clear in section 5, it is important that the operator components form a continuous linear program (LP) for the full model to be tractably solvable.

Operator sets and parameters

We define the operator sets for the different location types, commodities, bill of materials, transport modes and the network graph upon which the flow decisions are made. Operator parameters describe location capacities, system costs, transport mode characteristics, bill of material compositions, and customer demands. Table 4.1 outlines the sets, their indices, and the model parameters.

Table 4.1: Notations - Operator sets and parameters.

Set	Description
$i, j \in L$	Physical locations in the supply chain
$h \in L^S \subseteq L$	Supplier locations
$i \in L^P \subseteq L$	Producer locations
$j \in L^W \subseteq L$	Warehouse locations
$k \in L^C \subseteq L$	Customer locations
$(i, j) \in A$	Links between locations i and j
$p \in P$	Commodities, including raw materials and products
$b \in B$	Bills of Materials (BoM), specifying conversion processes
$p \in P(i, j)$	Commodities flowing on link (i, j)
$m \in M(i, j)$	Transport modes available on link (i, j)

Table 4.2: Notations - Parameters related to operations and costs.

Parameter	Description
$cap^S_{h,p}$	Maximum supply capacity of commodity p by supplier h
$cap^P_{i,b}$	Maximum production capacity of bill of material b by producer i
$cap^W_{j,p}$	Maximum storage capacity of commodity p at warehouse j
$dem^C_{k,p}$	Demand of commodity p by customer k
$g^{in}_{b,p}$	Input amount of commodity p required per production unit of bill of material b
$g^{out}_{b,p}$	Output amount of commodity p produced per production unit of bill of material b
$c^{fix,S}_h$	Fixed operating cost of using supplier h
$c^{fix,P}_i$	Fixed operating cost / rent of using producer i
$c^{fix,W}_j$	Fixed operating cost / rent of using warehouse j
$c^{pr,S}_{h,p}$	Cost of buying a unit of commodity p from supplier h
$c^{pr,P}_{i,b}$	Cost of producing a unit of bill of material b at producer i
$c^{pr,W}_{j,p}$	Cost of storing a unit of commodity p at warehouse j
$c^{rv,C}_{k,p}$	Penalty cost for failing to deliver a unit of commodity p to customer k
$c^{tr}_{i,j,m}$	Total trip cost between facilities i and j using mode m
c^{trip}_m	Fixed cost to start a trip using mode m
c^{dist}_m	Variable cost per unit of distance using mode m
$d_{i,j}$	Distance between two locations i and j
ls_m	Average load size of mode m
lc_p	Transport load unit conversion of commodity p

Operator decision variables

The first group of operational decision variables, $q^S_{h,p}$, $q^P_{i,b}$, $q^W_{j,p}$, are process variables. These variables quantify the amounts of commodities (raw materials) supplied, (bills of material) production processes produced, and commodities (final products) stored at warehouses. Variables $q^C_{k,p}$ and $\bar{q}^C_{k,p}$ represent the amount of final products delivered to the customer and the amount not delivered (i.e., the shortfall in meeting customer demand), respectively. See Table 4.3 for an overview of all the variables of the operator model. Decision variables $y_{i,j,p}$ represent the flow of commodities between different locations, in the commodity's unit. The final operational variables are the link transport loads, denoted as $z_{i,j,m}$. These variables represent the number of trips using transportation mode m between the various facilities. For the resolution approach, we require the operator variables to be continuous, even though some processes may be more realistically modelled as integers (e.g the number of transport trips). Continuous variables can however sufficiently approximate integer variables when the values are large enough and the appropriate feasibility measures are included. For a more concise notation, we define the vector of operational variables: $\boldsymbol{x_O} = \begin{bmatrix} \boldsymbol{q^S} & \boldsymbol{q^P} & \boldsymbol{q^W} & \boldsymbol{q^C} & \boldsymbol{\bar{q}^C} & \boldsymbol{y} & \boldsymbol{z} \end{bmatrix}$

Operator objective: system operational performance

Table 4.3: Notations - Operator decision variables $\boldsymbol{x_O}$

Operations x_O	
$q^S_{h,p}$	Amount of commodity p supplied from supplier h
$q^P_{i,b}$	Amount of bill of material b produced at producer i
$q^W_{j,p}$	Amount of commodity p stored at warehouse j
$q^C_{k,p}$	Amount of commodity p delivered to customer k
$\bar{q}^C_{k,p}$	Amount of commodity p *not* delivered to customer k (difference from demand)
$y_{i,j,p}$	Commodity flow - amount of commodity p between locations i and j
$z_{i,j,m}$	Transport load - number of trips between locations i and j using mode m

The operator aims to achieve optimal operational performance. The objective of the operator model is to minimize a cost function Γ^{total}, comprising two components: Γ^{oper} and Γ^{penal}. Γ^{oper} represents operational costs over the defined lifetime (e.g., a full year), while Γ^{penal} penalizes the non-delivery of customer demand, serving as a soft constraint to model resilience. These terms are weighted by user-chosen parameters ρ^c and ρ^r, where $\rho^c + \rho^r = 1$.

$$\begin{aligned} \underset{\boldsymbol{x_O}}{min} \quad \Gamma^{total} &= \rho^r \cdot \Gamma^{penal} + \rho^c \cdot \Gamma^{costs} \\ &= \rho^r \cdot \Gamma^{penal} + \rho^c \cdot (\Gamma^{proc} + \quad \Gamma^{trans}) \end{aligned} \tag{4.1}$$

Here, Γ^{oper} comprises process costs Γ^{proc} and transport costs Γ^{trans}. The **process costs** Γ^{proc} consist of $\Gamma^{proc,S}$, $\Gamma^{proc,P}$, and $\Gamma^{proc,W}$, representing costs related to resource utilization in the supply, production, and storage process steps. See 4.1a. **Transport costs** Γ^{trans} are calculated by summing the product of the transport load volumes per mode ($z^{\cdot}$), i.e. the number of trips of a certain mode between two facilities, and their associated trip cost coefficients ($c^{tr,\cdot}$), according to equation 4.1b.

$$\begin{aligned} \Gamma^{proc} &= \Gamma^{proc,S} + \Gamma^{proc,P} + \Gamma^{proc,W} \\ \Gamma^{proc,S} &= \sum_{h \in L^S} \sum_{p \in p} c^{pr,S}_{h,p} \cdot q^S_{h,p} \\ \Gamma^{proc,P} &= \sum_{i \in L^P} \sum_{b \in B} c^{pr,P}_{i,b} \cdot q^P_{i,b} \\ \Gamma^{proc,W} &= \sum_{j \in L^W} \sum_{p \in P} c^{pr,W}_{j,p} \cdot q^W_{j,p} \end{aligned} \tag{4.1a}$$

$$\Gamma^{trans} = \sum_{(i,j) \in A} \sum_{m \in M(i,j)} c^{tr}_{i,j,m} \cdot z_{i,j,m} = \sum_{(i,j) \in A} \sum_{m \in M(i,j)} (c^{trip}_m + c^{dist}_m \cdot d_{i,j}) \cdot z_{i,j,m} \tag{4.1b}$$

Finally, the non-delivery **penalty costs** Γ^{penal} are associated with penalties incurred for failing to deliver a commodity (final product) to a customer. We include the non-delivery of commodities as a penalty term to the objective, instead of including hard demand-satisfying constraints. This is such that the objective function can be used to explore supply chain resilience by allowing the amount of customer deliveries to be impacted by events. Equation 4.1c calculates these costs as the product of the amount of final products *not* delivered ($\bar{q}^C_{k,p}$) and the associated lost revenue ($c^{rv,C}_{k,p}$) over all customers and final products.

$$\Gamma^{penal} = \sum_{k \in L^C} \sum_{p \in P} c^{rv,C}_{k,p} \cdot \bar{q}^C_{k,p} \tag{4.1c}$$

Operator constraints

We define the facility capacity constraints 4.2, 4.3, and 4.4 which ensure that the amount of commodities (raw materials) supplied, bill of material processes produced or commodities (final products) stored at a facility does not exceed its capacity, respectively.

$$
\begin{aligned}
q^S_{h,p} &\leq cap^S_{h,p} && \forall h \in L^S, \quad \forall p \in P && (4.2)\\
q^P_{i,b} &\leq cap^P_{i,b} && \forall i \in L^P, \quad \forall b \in B && (4.3)\\
q^W_{j,p} &\leq cap^W_{h,p} && \forall j \in L^W, \quad \forall p \in P && (4.4)
\end{aligned}
$$

We then define various balance constraints that link the process variables to the incoming and outgoing commodity flows. The supply constraints 4.5 guarantee that the amount of raw materials supplied by a supplier is equal to the total outgoing flows of those commodities. The warehouse flow balance constraints 4.6 maintain the balance of commodities flowing into and out of warehouses. Constraints 4.7 link the amount of delivered final products to a customer to the sum of the incoming product flows.

$$
\begin{aligned}
q^S_{h,p} &= \sum_{i \in n^{out}(h,p)} y_{h,i,p} && && \forall h \in L^S, \quad \forall p \in P && (4.5)\\
q^W_{j,p} &= \sum_{i \in n^{in}(j,p)} y_{i,j,p} &&= \sum_{k \in n^{out}(j,p)} y_{j,k,p} && \forall j \in L^w, \quad \forall p \in P && (4.6)\\
q^C_{k,p} &= \sum_{j \in n^{in}(k,p)} y_{j,k,p} && && \forall k \in L^C, \quad \forall p \in P && (4.7)
\end{aligned}
$$

Where $n^{in}(i,p)$ returns all locations from which location i has a possible incoming flow of commodity p, and $n^{out}(i,p)$ returns all locations to which location i has a possible outgoing flow of commodity p. Furthermore, the production balance constraints, or bill of material (BOM) balance constraints 4.8 and 4.9 ensure that for each production process, there is a balance in input and output materials, based on the process' BOM definition. The customer demands constraints 4.10 ensure that the delivered and undelivered amounts of final products sum up to the customer demand of that product.

$$
\begin{aligned}
\sum_{b \in B} q^P_{i,b} \cdot g^{in}_{b,p} &\leq \sum_{h \in n^{in}(i,p)} y_{h,i,p} && \forall i \in L^P, \quad \forall p \in P && (4.8)\\
\sum_{b \in B} q^P_{i,b} \cdot g^{out}_{b,p} &\geq \sum_{j \in n^{out}(i,p)} y_{i,j,p} && \forall i \in L^P, \quad \forall p \in P && (4.9)\\
q^C_{k,p} + \bar{q}^C_{k,p} &= dem^C_{k,p} && \forall k \in L^C, \quad \forall p \in P && (4.10)
\end{aligned}
$$

The transportation constraints 4.11 ensure that, for each link between two facilities, a sufficient number of trips using the different transport modes are operated to carry the flow of commodities. The total number of trips using all available modes between two locations, multiplied by the mode's average trip load size, should be at least as large as the item flow between them. The commodity flow variables are multiplied by a load conversion parameter lc_p, in order to convert the flow, specified in the commodity's own unit, into a standardized transportation unit. These constraints assume that commodities transported between to specific locations may share a transport trip. This is a simplification that may not always hold in reality but is useful for now.

$$
\sum_{p \in P(i,j)} y_{i,j,p} \cdot ls_m \leq \sum_{m \in M(i,j)} z_{i,j,m} \cdot lc_p \qquad \forall (i,j) \in A \qquad (4.11)
$$

4.2.2. Incorporating the operational setting

In this subsection, we expand the previously introduced operator model by incorporating the notion of an operational setting. This setting describes events outside the operator's control that may affect the system's operations and performance. Disruptions of facilities or transport links, such as attacks by a disruptive agent, are encoded in an operational setting state, defining simultaneous events affecting the system. A setting state with no disruptive or unusual event is also considered, i.e., the "normal operations" setting.

Operational setting (attacker) sets and parameters

The following attacker set and parameters are now introduced. We model events through a disruptive agent making operational setting decisions in order to affect the system's operations. We introduce a total attack budget bdg^{att} and cost $c^{att,\cdot}$ parameters to model how a disruptive agent needs to allocate their resources to impact a system. This is relevant for intentional disruptive agents, but also random, natural disruptions, as it is unlikely that every facility in an entire system may be affected simultaneously. To model partial disruptions, we introduce a new set of disruption impact levels F that quantify varying degrees of perturbation to a process.

Table 4.4: Notations - Attacker set and parameters.

Sets	
F	Set of disruption impact levels f
Parameters	
bdg^{att}	Total available attack budget
$c_i^{att,loc}$	Attack cost of fully disabling facility i
$c_{h,p,f}^{att,pr,S}$	Attack cost of partially disabling the supply of commodity p at supplier h by impact level f
$c_{i,b,f}^{att,pr,P}$	Attack cost of partially disabling the production of bill of material b at producer i by impact level f
$c_{ijm}^{att,link}$	Attack cost of fully disabling transport mode m on link (i,j)
u_f	Impact (% disabled) of disruption impact level f

Operational setting decision variables

The following decision variables are introduced enabling the disruptive agent to make "choices" on how to affect the supply chain system, i.e. to cause (disruptive) events. All event decisions are binary: an event occurs, or it does not. Attack variables ϕ_h^S, ϕ_i^P and ϕ_j^W relate to whether a supplier, producer or warehouse location is completely disabled. This may reflect real life events such as flooding or supplier bankruptcy that can partially or fully disrupt the operations of a location. $\psi_{h,p,f}^S$ and $\psi_{i,b,f}^P$ affect whether a particular process at a facility, i.e. the supply of a certain commodity from a supplier or the production of a certain bill of material process, is disrupted with a disruption impact level f. Finally, variables $\tau_{i,j,m}$, affect whether a particular transport mode m on a particular link (i,j) is fully disabled. This could represent a real-life disruption to a shipping lane or railway link, for example. We could also define a commodity-specific storage disruption at a warehouse $\psi_{j,p,f}^W$, but this has been omitted in our implementation. For a more concise notation, we define the operational setting (attacks) variables vector: $\boldsymbol{x_A} = \begin{bmatrix}\boldsymbol{\phi^S} & \boldsymbol{\phi^P} & \boldsymbol{\phi^W} & \boldsymbol{\psi^S} & \boldsymbol{\psi^P} & \boldsymbol{\tau}\end{bmatrix}$

Table 4.5: Notations - Attacker decision variables $\boldsymbol{x_A}$

Attack x_A	
ϕ_h^S	Binary variable, 1 if supplier location h is fully disabled
ϕ_i^P	Binary variable, 1 if producer location i is fully disabled
ϕ_j^W	Binary variable, 1 if warehouse location j is fully disabled
$\psi_{h,p,f}^S$	Binary variable, 1 if the supply of commodity p from supplier h is disrupted by level f
$\psi_{i,b,f}^P$	Binary variable, 1 if the production of bill of material b at producer i is disrupted by level f
$\tau_{i,j,m}$	Binary variable, 1 if transport mode m between locations i and j is fully disabled

Operational setting objective function

With the attacker component introduced, we can turn the model into a bi-level program. This entails modifying the objective to include the second-level optimisation operator. The objective function as redefined below, aims to optimise for worst performance, i.e. maximize the minimal cost function value.

$$\underset{\boldsymbol{x_A}}{max}\ \underset{\boldsymbol{x_O}}{min} \quad \Gamma^{total} = \quad \rho^r \cdot \Gamma^{penal} + \rho^c \cdot (\Gamma^{proc} + \quad \Gamma^{trans}) \tag{4.1}$$

Operational setting constraints

A new constraint relating to the resources allocation of the attacker should be included. It relates the costs of performing various attacks to the attack budget parameter. The total attack costs include costs for fully disabling locations $\Gamma^{att,loc}$, partially disrupting supply and production processes $\Gamma^{att,proc}$, and disabling transport links $\Gamma^{att,trans}$, and are defined as follows.

$$\Gamma^{attacks} \leq bdg^{att} \tag{4.12}$$

$$\begin{aligned}
\Gamma^{attacks} &= \Gamma^{att,loc} + \Gamma^{att,proc} + \Gamma^{att,trans} \qquad (4.12a)\\
\Gamma^{att,loc} &= \sum_{h\in L^S} c_h^{att,loc}\cdot\phi_h^S + \sum_{i\in L^P} c_i^{att,loc}\cdot\phi_i^P + \sum_{j\in L^W} c_j^{att,loc}\cdot\phi_j^W\\
\Gamma^{att,proc} &= \sum_{f\in F}\Big(\sum_{h\in L^S}\sum_{p\in P} c_{h,p,f}^{att,pr,S}\cdot\psi_{h,p,f}^S + \sum_{i\in L^P}\sum_{b\in B} c_{i,b,f}^{att,pr,P}\cdot\psi_{i,b,f}^P\Big)\\
\Gamma^{att,trans} &= \sum_{(i,j)\in A}\sum_{m\in M(i,j)} c_{i,j,m}^{att,trans}\cdot\tau_{i,j,m}
\end{aligned}$$

Finally, the previously defined facility capacity constraints 4.2 - 4.4 should be modified and split into 4.2a - 4.4a and 4.2b - 4.3b to account for the location and process-specific attacks, respectively. These constraint modifications ensure that the attack variables have an actual effect on the operational capacities of the facilities. The transportation constraints 4.11 are also modified in a similar way.

$$\begin{aligned}
q_{h,p}^S &\leq cap_{h,p}^S \quad (1-\phi_h^S) && \forall h\in L^S,\quad \forall p\in P && (4.2a)\\
q_{i,b}^P &\leq cap_{i,b}^P \quad (1-\phi_i^P) && \forall i\in L^P,\quad \forall b\in B && (4.3a)\\
q_{j,p}^W &\leq cap_{h,p}^W \quad (1-\phi_j^W) && \forall j\in L^W,\quad \forall p\in P && (4.4a)\\
q_{h,p}^S &\leq cap_{h,p}^S \quad (1-u_f\cdot\psi_{h,p,f}^S) && \forall h\in L^S,\quad \forall p\in P,\quad \forall f\in F && (4.2b)\\
q_{i,b}^P &\leq cap_{i,b}^P \quad (1-u_f\cdot\psi_{i,b,f}^P) && \forall i\in L^P,\quad \forall b\in B,\quad \forall f\in F && (4.3b)\\
\sum_{p\in P(i,j)} y_{i,j,p}\cdot ls_m &\leq \sum_{m\in M(i,j)} z_{i,j,m}\cdot lc_p\cdot(1-\tau_{i,j,m}) && \forall (i,j)\in A && (4.11)
\end{aligned}$$

4.2.3. Incorporating system design and defence decisions

The third optimisation level will now be introduced to complete the tri-level model. This decision level involves strategic decisions regarding the design of the system. As the system design will aim to be resilient against various, possibly disruptive operational settings, this decision level can also be referred to as the defence layer. These decisions allow the operator or designer to invest some initial budget in the initiation or modification of the strategic design of the system in anticipation of various operational settings. In our case, the design decisions mainly regard the supply chain network layout, i.e. the used facility locations.

System design parameters

Similar to the parameters introduced in the attacker's component, the designer parameters consist of a certain total initial investment budget bdg^{def} and investment costs for using / opening a location $c_i^{init,\cdot}$. In the real world, the initial cost of using a supplier may consist of administrative costs incurred by forming a deal with that supplier. Producer and warehouse initial costs may involve building, renovation, or land purchase costs. These costs could be set to zero if the location already exists within the system.

Table 4.6: Notations - System design parameters.

Parameters	
bdg^{def}	Total available system design / defence budget
$c_h^{init,S}$	Initial investment cost of using supplier h
$c_i^{init,P}$	Initial investment cost of using producer i
$c_j^{init,W}$	Initial investment cost of using warehouse j

System design decision variables

The design variables involve binary decisions x_i regarding the facility locations to be used. In this model, no particular distinction is made between locations that already exist within a supply chain system, and those that are candidates to be newly opened. Similarly, suppliers are generally external parties but the decision to use one is modelled in the same way as any other facility location decision. At this step, for a particular location one could introduce different levels of investment. For simplicity this is not included for now. For a more concise notation, we define the system design variables vector: $\boldsymbol{x_D} = \begin{bmatrix} \boldsymbol{x^S} & \boldsymbol{x^P} & \boldsymbol{x^W} \end{bmatrix}$

Table 4.7: Notations - System design decision variables $\boldsymbol{x_D}$

System design x_D	
x_h^S	Binary variable, 1 if supplier location h is used
x_i^P	Binary variable, 1 if producer location i is used
x_j^W	Binary variable, 1 if warehouse location j is used

System design objective function

At this step, an extra term representing the **fixed operational costs** Γ^{fixed} is added to the performance function and represents the fixed costs incurred from using a facility. These costs may for example include rent, administrative, energy costs, and are calculated, according to equation 4.1d. These costs are *not* to be confused with the initial investment costs, which are one-time strategic costs.

$$\begin{aligned}
\Gamma^{fixed} &= \Gamma^{fixed,S} + \Gamma^{fixed,P} + \Gamma^{fixed,W} \\
\Gamma^{fixed,S} &= \sum_{h \in L^S} c_h^{fix,S} \cdot x_h^S \\
\Gamma^{fixed,P} &= \sum_{i \in L^P} c_i^{fix,P} \cdot x_i^P \\
\Gamma^{fixed,W} &= \sum_{j \in L^W} c_j^{fix,W} \cdot x_j^W
\end{aligned} \tag{4.1d}$$

The third optimisation level is added to the objective function of the full tri-level model. The full model's objective is to minimize the operational costs including penalties, assuming an optimal (minimized) operational response to the most disruptive operational setting - i.e. $min - max - min$ total costs. The components of the objective function are summarize in table 4.8.

$$\min_{\boldsymbol{x_D}} \max_{\boldsymbol{x_A}} \min_{\boldsymbol{x_O}} \quad \Gamma^{total} = \rho^r \cdot \Gamma^{penal} + \rho^c \cdot (\Gamma^{fixed} + \Gamma^{proc} + \Gamma^{trans}) \tag{4.1}$$

Table 4.8: Overview of objective function components

Costs	
Γ^{total}	Performance (cost) function
Γ^{costs}	Operational costs
Γ^{penal}	Penalty cost term for non-delivery of demand
Γ^{fixed}	Total fixed operational facility use costs
$\Gamma^{fixed,S}$	Fixed operational supplier use costs
$\Gamma^{fixed,P}$	Fixed operational producer use costs
$\Gamma^{fixed,W}$	Fixed operational warehouse use costs
Γ^{proc}	Total process costs
$\Gamma^{proc,S}$	Commodity (raw materials) supply costs
$\Gamma^{proc,P}$	Bill of material production costs
$\Gamma^{proc,W}$	Commodity (final product) storage costs
Γ^{trans}	Transportation costs

System design constraints

A new constraint 4.13 is added to model the designer's resources allocation, relating the initial design investment budget bdg^{def} to the design costs Γ^{design}.

$$\Gamma^{design} = \sum_{h \in L^S} c_h^{init,S} \cdot x_h^S + \sum_{i \in L^P} c_i^{init,P} \cdot x_i^P + \sum_{j \in L^W} c^{init,W} \cdot x_j^W \leq bdg^{def} \tag{4.13}$$

New facility capacity constraints 4.2c - 4.4c should be added to ensure that a facility's capacity can only be used if that facility is actually included in the system design.

$$q_{h,p}^S \leq cap_{h,p}^S \cdot x_h^S \qquad \forall h \in L^S, \quad \forall p \in P \tag{4.2c}$$
$$q_{i,b}^P \leq cap_{i,b}^P \cdot x_i^P \qquad \forall i \in L^P, \quad \forall b \in B \tag{4.3c}$$
$$q_{j,p}^W \leq cap_{h,p}^W \cdot x_j^W \qquad \forall j \in L^W, \quad \forall p \in P \tag{4.4c}$$

As there are now multiple capacity constraints defined per location (4.2a - 4.4c), a more concise notation using the *min* function can be introduced to redefine (4.2 - 4.4):

$$q_{h,p}^S \leq cap_{h,p}^S \cdot min(x_h^S; \quad 1 - \phi_h^S; \quad 1 - u_f \cdot \psi_{h,p,f}^S \quad \forall f \in F) \qquad \forall h \in L^S, \quad \forall p \in P \tag{4.2}$$
$$q_{i,b}^P \leq cap_{i,b}^P \cdot min(x_i^P; \quad 1 - \phi_i^P; \quad 1 - u_f \cdot \psi_{i,b,f}^P \quad \forall f \in F) \qquad \forall i \in L^P, \quad \forall b \in B \tag{4.3}$$
$$q_{j,p}^W \leq cap_{h,p}^W \cdot min(x_j^W; \quad 1 - \phi_j^W) \qquad \forall j \in L^W, \quad \forall p \in P \tag{4.4}$$

4.3. Full Operator-Attacker-Defender model: mOAD

Finally, the full tri-level model is written down as follows. The variable domain constraints (vector form) have been defined in 4.14 - 4.16 and the capacity constraints 4.2 - 4.4 have been rewritten using the *min* function for brevity. We will name this model OAD model adapted to the operation of a supply chain **mOAD**. It should be obvious that the different levels of decisions can be adapted according to the circumstances. Different operational variables could be defined if the operations involve other activities. Likewise, different disruptive events or system design / defence choices could be introduced. In any case, the three levels of intertwined decisions cause this model to be difficult to solve using classic approaches. A resolution strategy is required to solve it and this will be discussed in detail in the next chapter.

[**mOAD**]

$$\min_{\boldsymbol{x_D}} \max_{\boldsymbol{x_A}} \min_{\boldsymbol{x_O}} \quad \Gamma^{total} = \quad \rho^r \cdot \Gamma^{penal} + \rho^c \cdot (\Gamma^{fixed} + \Gamma^{proc} + \Gamma^{trans}) \tag{4.1}$$

s.t.

$$q^S_{h,p} \leq cap^S_{h,p} \cdot min(x^S_h; 1 - \phi^S_h; 1 - u_f \cdot \psi^S_{h,p,f} \forall f \in F) \qquad \forall h \in L^S, \forall p \in P \tag{4.2}$$

$$q^P_{i,b} \leq cap^P_{i,b} \cdot min(x^P_i; 1 - \phi^P_i; 1 - u_f \cdot \psi^P_{i,b,f} \forall f \in F) \qquad \forall i \in L^P, \forall b \in B \tag{4.3}$$

$$q^W_{j,p} \leq cap^W_{h,p} \cdot min(x^W_j; 1 - \phi^W_j) \qquad \forall j \in L^W, \forall p \in P \tag{4.4}$$

$$q^S_{h,p} = \sum_{i \in n^{out}(h,p)} y_{h,i,p} \qquad \forall h \in L^S, \forall p \in P \tag{4.5}$$

$$q^W_{j,p} = \sum_{i \in n^{in}(j,p)} y_{i,j,p} = \sum_{k \in n^{out}(j,p)} y_{j,k,p} \qquad \forall j \in L^w, \forall p \in P \tag{4.6}$$

$$q^C_{k,p} = \sum_{j \in n^{in}(k,p)} y_{j,k,p} \qquad \forall k \in L^C, \forall p \in P \tag{4.7}$$

$$\sum_{b \in B} q^P_{i,b} \cdot g^{in}_{b,p} \leq \sum_{h \in n^{in}(i,p)} y_{h,i,p} \qquad \forall i \in L^P, \forall p \in P \tag{4.8}$$

$$\sum_{b \in B} q^P_{i,b} \cdot g^{out}_{b,p} \geq \sum_{j \in n^{out}(i,p)} y_{i,j,p} \qquad \forall i \in L^P, \forall p \in P \tag{4.9}$$

$$q^C_{k,p} + \bar{q}^C_{k,p} = dem^C_{k,p} \qquad \forall k \in L^W, \forall p \in P \tag{4.10}$$

$$\sum_{p \in P(i,j)} y_{i,j,p} \cdot ls_m \leq \sum_{m \in M(i,j)} z_{i,j,m} \cdot lc_p \cdot (1 - \tau_{i,j,m}) \qquad \forall (i,j) \in A \tag{4.11}$$

$$\Gamma^{attacks} \leq bdg^{att} \tag{4.12}$$

$$\Gamma^{design} \leq bdg^{def} \tag{4.13}$$

$$\mathbf{y}, \mathbf{z}, \mathbf{q}, \bar{\mathbf{q}} \geq \mathbf{0} \tag{4.14}$$

$$\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\tau} \in \{0,1\} \tag{4.15}$$

$$\mathbf{x} \in \{0,1\} \tag{4.16}$$

5

Resolution strategy

The tri-level **mOAD** model with hierarchical decision-making cannot trivially be solved without a resolution strategy. In this section, we present a decomposition approach inspired by the methods of alderson2011solving and ghorbani2021decomposition. The core idea of our approach is to decompose the tri-level **mOAD** model into two parametrized models: a single-level master problem (**mOAD-MASTER**), which provides a lower bound, and a bi-level attacker sub-problem (**mOA-SUB**), which determines the upper bound on the solution of the tri-level optimization.

First, in 5.1, a family of parameterized models are introduced, where some of the decision variables of **mOAD** are fixed as given parameters. Then, in 5.2, a method is introduced for decomposing the tri-level **mOAD** problem into a relaxed master problem **mOAD-MASTER** and an attacker sub-problem **mOA-SUB**, a solution for which is provided in 5.3.

5.1. Parameterized models

From **mOAD**, we can create a family of parameterized models by fixing some of the decision variables as given parameters. As such, we will define the following parameterized models: 1. **mO**, 2. **mOA**, and 3. **mOD**.

1. **mO** $(\hat{x_A}, \hat{x_D}) \rightarrow {x_O}^*$ (the base operator model): its solution provides the optimal operational response ${x_O}^*$ for a given fixed system design $\hat{x_D}$ and fixed operational setting / attack plan $\hat{x_A}$. It is

defined below. Note that for a fixed design $\hat{\boldsymbol{x_D}}$, the $\Gamma^{fixed}(\hat{\boldsymbol{x_D}})$ term is a constant.

[**mO**]

$$\min_{\boldsymbol{x_O}} \quad \Gamma^{total} \quad (\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) = \quad \rho^r \cdot \Gamma^{penal} + \rho^c \cdot (\Gamma^{fixed}(\hat{\boldsymbol{x_D}}) \quad + \Gamma^{proc} + \Gamma^{trans}) \tag{4.1}$$

s.t. (5.1)

$$q^S_{h,p} \leq cap^S_{h,p} \cdot min(\hat{x^S_h}; \quad 1 - \hat{\phi^S_h}; \quad 1 - u_f \cdot \hat{\psi^S_{h,p,f}} \quad \forall f \in F) \qquad \forall h \in L^S, \quad \forall p \in P \tag{4.2}$$

$$q^P_{i,b} \leq cap^P_{i,b} \cdot min(\hat{x^P_i}; \quad 1 - \hat{\phi^P_i}; \quad 1 - u_f \cdot \hat{\psi^P_{i,b,f}} \quad \forall f \in F) \qquad \forall i \in L^P, \quad \forall b \in B \tag{4.3}$$

$$q^W_{j,p} \leq cap^S_{h,p} \cdot min(\hat{x^W_j}; \quad 1 - \hat{\phi^W_j}) \qquad \forall j \in L^W, \quad \forall p \in P \tag{4.4}$$

$$q^S_{h,p} = \sum_{i \in n^{out}(h,p)} y_{h,i,p} \qquad \forall h \in L^S, \quad \forall p \in P \tag{4.5}$$

$$q^W_{j,p} = \sum_{i \in n^{in}(j,p)} y_{i,j,p} = \sum_{k \in n^{out}(j,p)} y_{j,k,p} \qquad \forall j \in L^w, \quad \forall p \in P \tag{4.6}$$

$$q^C_{k,p} = \sum_{j \in n^{in}(k,p)} y_{j,k,p} \qquad \forall k \in L^c, \quad \forall p \in P \tag{4.7}$$

$$\sum_{b \in B} q^P_{i,b} \cdot g^{in}_{b,p} \leq \sum_{h \in n^{in}(i,p)} y_{h,i,p} \qquad \forall i \in L^P, \quad \forall p \in P \tag{4.8}$$

$$\sum_{b \in B} q^P_{i,b} \cdot g^{out}_{b,p} \geq \sum_{j \in n^{out}(i,p)} y_{i,j,p} \qquad \forall i \in L^P, \quad \forall p \in P \tag{4.9}$$

$$q^C_{k,p} + \bar{q}^C_{k,p} = dem^C_{k,p} \qquad \forall k \in L^W, \quad \forall p \in P \tag{4.10}$$

$$\sum_{p \in P(i,j)} y_{i,j,p} \cdot ls_m \leq \sum_{m \in M(i,j)} z_{i,j,m} \cdot lc_p \cdot (1 - \tau_{i,\hat{j},m}) \qquad \forall (i,j) \in A \tag{4.11}$$

$$\mathbf{y}, \quad \mathbf{z}, \quad \mathbf{q}, \quad \bar{\mathbf{q}} \geq \quad \mathbf{0} \tag{4.14}$$

2. **mOA** $(\hat{\boldsymbol{x_D}}) \rightarrow \boldsymbol{x_O}^*, \boldsymbol{x_A}^*$ (the bi-level attacker model): its solution provides the "optimal" attack plan $\boldsymbol{x_A}^*$ (from the interdictor's perspective) for a given fixed system design $\hat{\boldsymbol{x_D}}$ to cause as much harm to the operational performance of the system. Of course, it assumes that the operator will react optimally to the operational setting.

[**mOA**]

$$\max_{\boldsymbol{x_A}} \min_{\boldsymbol{x_O}} \quad \Gamma^{total} \quad (\hat{\boldsymbol{x_D}}) \quad = \quad \rho^r \cdot \Gamma^{penal} + \rho^c \cdot (\Gamma^{fixed}(\hat{\boldsymbol{x_D}}) \quad + \Gamma^{proc} + \Gamma^{trans}) \tag{4.1}$$

s.t. Operator constraints

(4.2) - (4.11)

$$\Gamma^{attacks} \leq bdg^{att} \tag{4.12}$$

$$\mathbf{y}, \quad \mathbf{z}, \quad \mathbf{q}, \quad \bar{\mathbf{q}} \geq \quad \mathbf{0} \tag{4.14}$$

$$\boldsymbol{\phi}, \quad \boldsymbol{\psi}, \quad \boldsymbol{\tau} \quad \in \quad \{0,1\} \tag{4.15}$$

3. **mOD** $(\hat{\boldsymbol{x_A}}) \rightarrow \boldsymbol{x_O}^*, \boldsymbol{x_D}^*$ (the bi-level designer model): its solution provides the optimal system design $\boldsymbol{x_D}^*$ for a given operational setting $\hat{\boldsymbol{x_A}}$. This model may not directly have a realistic implementation in the real world as the previous ones do, as one cannot predict the operational setting before a system is designed. It will however still be useful for solving **mOAD** later on.

[**mOD**]

$$\min_{\boldsymbol{x_O}, \boldsymbol{x_D}} \quad \Gamma^{total} \quad (\hat{\boldsymbol{x_A}}) \quad = \quad \rho^r \cdot \Gamma^{penal} + \rho^c \cdot (\Gamma^{fixed} + \Gamma^{proc} + \Gamma^{trans}) \tag{4.1}$$

s.t. Operator constraints

(4.2) - (4.11)

$$\Gamma^{design} \leq bdg^{def} \tag{4.13}$$

$$\mathbf{y}, \quad \mathbf{z}, \quad \mathbf{q}, \quad \bar{\mathbf{q}} \geq \quad \mathbf{0} \tag{4.14}$$

$$\boldsymbol{\phi}, \quad \boldsymbol{\psi}, \quad \boldsymbol{\tau} \quad \in \quad \{0,1\} \tag{4.15}$$

$$\mathbf{x} \quad \in \quad \{0,1\} \tag{4.16}$$

5.2. mOAD decomposition

One can notice that for a given operational setting k: $\hat{\boldsymbol{x}_{\boldsymbol{Ak}}}$, there exists a corresponding operational response $\hat{\boldsymbol{x}_{\boldsymbol{Ok}}}$ forming a pair $(\hat{\boldsymbol{x}_{\boldsymbol{Ak}}}, \hat{\boldsymbol{x}_{\boldsymbol{Ok}}})$. Using this, we can decompose the **mOAD** into a relaxed master problem **mOAD-MASTER** and an attacker sub-problem **mOA-SUB**. **mOAD-MASTER** returns an optimal system design $\boldsymbol{x_D}^*$ given a set of possible attacks plans, and the associated optimal attack-response pair $(\boldsymbol{x_A}^*, \boldsymbol{x_O}^*)$. **mOA-SUB** returns an optimal attack plan $\boldsymbol{x_A}^*$, given a system design $\hat{\boldsymbol{x}_{\boldsymbol{D}}}$. One could enumerate all possible attack plans give a system design and solve the master problem, but this set would be too large. Instead, we can create a subset of possible attack plans $\hat{\boldsymbol{X}_{\boldsymbol{A}}^{\boldsymbol{K}}} = \{\hat{\boldsymbol{x}_{\boldsymbol{A1}}}, ..., \hat{\boldsymbol{x}_{\boldsymbol{Ak}}}\}$. The key to the solution approach is to iteratively solve the master and sub-problems: solve the attacker sub-problem to update the set of possible attack plans, and then re-solve the master problem for the optimal design. This process is repeated until a stopping criteria is met.

5.2.1. Iterative resolution algorithm

The procedure described above is turned into a more precisely defined iterative algorithm, the pseudo-code of which is presented in 1. The algorithm initialisation requires the specification of the full input data that **mOAD** requires. It furthermore requires that a certain optimality gap ε where $0 \leq \varepsilon < 1$ and a maximum number of iterations K_{MAX} be defined. An optimality gap larger than zero (but possible very small) is advised to get the algorithm to terminate reliably. At each iteration, the attacker sub-problem may update the upper bound to the optimal operational costs z^* (z^{UP}) and the master problem provides a lower bound (z^{LO}). If the normalized difference between these bounds is less than ε, the optimality condition is said to be met.

Algorithm 1 | mOAD iterative resolution

Input: Full **mOAD** input data, optimality tolerance ε, maximum iterations K_{MAX}
Output: $(\boldsymbol{x_D}^*, \boldsymbol{x_A}^*, \boldsymbol{x_O}^*)$: optimal solutions

Initialize an empty set of attack vectors $\hat{\boldsymbol{X}_{\boldsymbol{A}}^{\boldsymbol{0}}} \leftarrow \{\emptyset\}$
Select an initial feasible operational setting $\hat{\boldsymbol{x}_{\boldsymbol{A0}}}$ (e.g., ”no attack / normal setting”)
Solve **mOD**($\hat{\boldsymbol{x}_{\boldsymbol{A0}}}$) for optimal $\boldsymbol{x}_{\boldsymbol{D}1}$ and z^*
Initialize $z^{LO} \leftarrow z^*, z^{UP} \leftarrow +\infty, K \leftarrow 1$
while $z^{UP} - z^{LO} > |z^{LO}| \cdot \varepsilon$ *and* $K < K_{MAX}$ **do**
  Solve **mOA-SUB**($\hat{\boldsymbol{x}_{\boldsymbol{D}K}}$) for $\boldsymbol{x}_{\boldsymbol{A}K}$ and z
  if $z < z^{UP}$ **then**
    $\boldsymbol{x_D}^* \leftarrow \hat{\boldsymbol{x}_{\boldsymbol{D}K}}$; $\boldsymbol{x_A}^* \leftarrow \boldsymbol{x}_{\boldsymbol{A}K}$; $z^{UP} \leftarrow z$
  end
  Add $\boldsymbol{x}_{\boldsymbol{A}K}$ to the set of attacks: $\hat{\boldsymbol{X}_{\boldsymbol{A}}^{\boldsymbol{K}}} \leftarrow \hat{\boldsymbol{X}_{\boldsymbol{A}}^{\boldsymbol{K}}} \cup \{\boldsymbol{x}_{\boldsymbol{A}K}\}$
  Solve **mOAD-MASTER**($\hat{\boldsymbol{X}_{\boldsymbol{A}}^{\boldsymbol{K}}}$) for $\boldsymbol{x}_{\boldsymbol{D}K+1}$ and z
  if $z > z^{LO}$ **then**
    $\boldsymbol{x_D}^* \leftarrow \boldsymbol{x}_{\boldsymbol{D}K+1}$; $z^{LO} \leftarrow z$
  end
  $K \leftarrow K + 1$
end
Solve **mO**($\boldsymbol{x_D}^*, \boldsymbol{x_A}^*$) for $\boldsymbol{x_O}^*$
return $(\boldsymbol{x_D}^*, \boldsymbol{x_A}^*, \boldsymbol{x_O}^*)$

5.2.2. Master problem: mOAD-MASTER

The master problem *mOAD-MASTER* optimizes the system design $\boldsymbol{x_D}^*$ given a subset of possible disruption vectors $\hat{\boldsymbol{x}_{\boldsymbol{A}}^{\boldsymbol{K}}}$. It takes as input a subset of all possible attack plans $\hat{\boldsymbol{X}_{\boldsymbol{A}}^{\boldsymbol{K}}}$ and optimises for the optimal system design $\boldsymbol{x_D}^*$, i.e. the design that minimizes the operational cost function given the attacks that may happen from the set (the attack vectors $\hat{\boldsymbol{x}_{\boldsymbol{A}k}}$ are thus parameters in **mOAD-MASTER**, and not decision variables). Apart from the system design variables, the other decision variables that

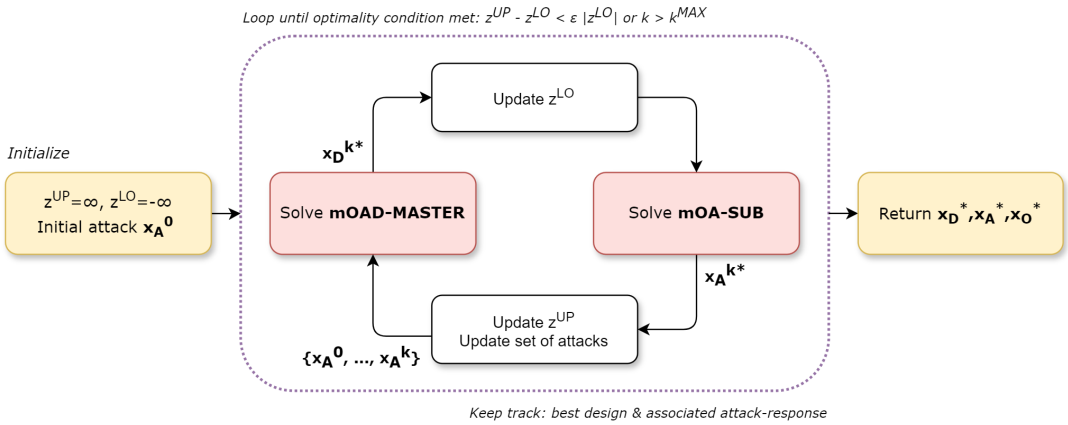


Figure 5.1: Overview of **mOAD** resolution decomposition procedure. The master problem and the attacker sub-problem are solved iteratively in a loop until the end criteria is met.

are included in **mOAD-MASTER** are Z, a single helper variable representing the total operational cost, and the operational variables $\boldsymbol{x_O}_1, ..., \boldsymbol{x_O}_K$ associated with each operational setting / attack vector in $\hat{\boldsymbol{X}}_{\boldsymbol{A}}^{\boldsymbol{K}}$. The design variables need to satisfy to the designer constraints. In our case the set of designer constraints essentially boil down to satisfying to the design/defence budget constraint 4.13. In the program definition below, we will write this more generally and concisely using constraint 5.3, where X_D denotes the set of all possible valid/feasible design vectors $\boldsymbol{x_D}$. Similarly, the operational variables $\boldsymbol{x_O}_1, ..., \boldsymbol{x_O}_K$ associated with each attack vector $\hat{\boldsymbol{x_A}}_1, ..., \hat{\boldsymbol{x_A}}_K$ need to satisfy to all the operational constraints 4.2 - 4.11. This implies that for each attack vector in $\hat{\boldsymbol{X}}_{\boldsymbol{A}}^{\boldsymbol{K}}$, all the operator constraints need to be added again, with the appropriate $\boldsymbol{x_D}$ and $\hat{\boldsymbol{x_A}}_k$. This is notated concisely using constraints 5.4, where $X_O(\boldsymbol{x_D}, \hat{\boldsymbol{x_A}}_k)$ is the set of valid operational variables for design $\boldsymbol{x_D}$ and setting $\hat{\boldsymbol{x_A}}_k$. The number of variables and constraints grows rapidly for each addition of $\hat{\boldsymbol{x_A}}_k$ to $\hat{\boldsymbol{X}}_{\boldsymbol{A}}^{\boldsymbol{K}}$. Finally, constraints 5.5 ensure that the objective value of the master problem is not smaller than the operational cost associated with each attack-operational response pair $(\hat{\boldsymbol{x_A}}_k, \hat{\boldsymbol{x_O}}_k)$.

[**mOAD-MASTER**]

$$\min_{Z, \boldsymbol{x_D}, \boldsymbol{x_O}_1, ..., \boldsymbol{x_O}_K} \quad Z \tag{5.2}$$

$$\text{s.t.} \quad \boldsymbol{x_D} \in X_D \tag{5.3}$$

$$\boldsymbol{x_O}_k \in X_O \quad (\boldsymbol{x_D}, \hat{\boldsymbol{x_A}}_k) \qquad \forall k \in K \tag{5.4}$$

$$Z \geq \Gamma^{total}(\boldsymbol{x_D}, \hat{\boldsymbol{x_A}}_k, \boldsymbol{x_O}_k) \qquad \forall k \in K \tag{5.5}$$

5.3. Resolution of the attacker sub-problem: mOA-SUB

The decomposition algorithm described above implies that a method exists to solve the bi-level attacker sub-problem **mOA-SUB**. A solution to the sub-prblem involving duality theory is proposed in this sub-section. The use of the strong duality theorem in this solution implies that the base operator model **mO** needs to be a continuous linear problem (LP). If the operator model was not an LP, solving the sub-problem would require another decomposition, which would seriously affect the tractability of the resolution.

5.3.1. Resolution approach overview

We employ a ”dualize-and-combine approach”. The first step in our approach to solve **mOA-SUB** is to find the dual linear problem, **mO-DUAL**, of the base operator model **mO**. According to the strong duality theorem, if **mO** represents a feasible linear program with an optimal solution, its dual counterpart, **mO-DUAL**, will also possess an optimal solution, with equivalent objective function values. Given that **mO** is aimed at minimization, its dual, **mO-DUAL**, naturally is a maximization problem. This shift allows for the the bi-level sub-problem **mOA-SUB** to be collapsed into a single-level quadratic problem dubbed **mOA-MIQP**. With a predetermined system design vector $\boldsymbol{x_D}$, solving **mOA-MIQP** provides insight into the ”optimal” attack plan $\boldsymbol{x_A}^*$ — identifying attacks that inflict the most damage on the operator’s performance function within the given system design. While solving the dual problem does not directly yield solutions to its primal counterpart, solving **mO** with the designated system design $\boldsymbol{x_D}$ and the identified optimal attack plan $\boldsymbol{x_A}^*$ allows for deriving the corresponding optimal operational response $\boldsymbol{x_O}^*$. As such, we can solve the sub-problem by redefining **mOA-SUB** := **mOA-MIQP**.

5.3.2. Dualization of the operator model: mO-DUAL

The linear operator program **mO** has a dual problem **mO-DUAL**. Let $\boldsymbol{x_\Omega}$ be the vector of dual operator variables, we can define mO-DUAL as follows:

[**mO-DUAL**]

$$\max_{\boldsymbol{x_\Omega}} \quad \Delta^{oper,dual} \quad (\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) \tag{5.15}$$

$$\text{s.t.} \quad \upsilon^S_{h,p} + \nu^S_{h,p} + \sum_{f\in F} \gamma^S_{h,p,f} - \alpha^S_{h,p} \quad \leq \quad c^{pr,S}_{h,p} \cdot \rho^c \qquad h \in L^S, \quad \forall p \in P \tag{5.6}$$

$$\upsilon^P_{i,b} + \nu^P_{i,b} + \sum_{f\in F} \gamma^P_{i,b,f} - \sum_{p\in P}(g^{in}_{b,p} \cdot b^{in}_{i,p} - g^{out}_{b,p} \cdot b^{out}_{i,p}) \quad \leq \quad c^{pr,P}_{i,b} \cdot \rho^c \qquad i \in L^P, \quad \forall b \in B \tag{5.7}$$

$$\upsilon^W_{j,p} + \nu^W_{j,p} - \alpha^{W,in}_{j,p} - \alpha^{W,out}_{j,p} \quad \leq \quad c^{pr,W}_{j,p} \cdot \rho^c \qquad j \in L^W, \quad \forall p \in P \tag{5.8}$$

$$\delta^C_{k,p} - \alpha^C_{k,p} \quad \leq \quad 0 \qquad k \in L^C, \quad \forall p \in P \tag{5.9}$$

$$\delta^C_{k,p} \quad \leq \quad c^{rv,C}_{k,p} \cdot \rho^r \qquad k \in L^C, \quad \forall p \in P \tag{5.10}$$

$$\alpha^S_{i,p} + \alpha^{W,in}_{j,p} + \alpha^{W,out}_{i,p} + \alpha^C_{j,p} + \beta^{in}_{i,p} + \beta^{out}_{i,p} + lc_p \cdot \mu_{i,j} \leq \quad 0 \qquad (i,j) \in A, \quad \forall p \in P(i,j) \tag{5.11}$$

$$-\, ls_m \cdot (1 - \tau_{i,\hat{j},m}) \cdot \mu_{i,j} \leq \quad c^{tr}_{i,j,m} \cdot \rho^c \qquad (i,j) \in A, \quad \forall m \in M(i,j) \tag{5.12}$$

$$\boldsymbol{\upsilon}, \quad \boldsymbol{\nu}, \quad \boldsymbol{\gamma}, \quad \boldsymbol{\beta}, \quad \boldsymbol{\mu} \leq \quad \mathbf{0} \tag{5.13}$$

$$\boldsymbol{\alpha}, \quad \boldsymbol{\delta} \in \quad \mathbb{R} \tag{5.14}$$

Each constraint in the primal is associated with a decision variable in the dual, an overview of which is found in table 5.1. Each variable in the primal is associated with a constraint in the dual, see the **mO-DUAL** definition and table 5.2 below. The objective of **mO-DUAL** is defined as in 5.15 and reuses the constant $\Gamma^{fixed}(\hat{\boldsymbol{x_D}})$ term from **mO**.

$$\begin{aligned}
\Delta^{oper,dual} \quad (\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) &= \Gamma^{fixed}(\hat{\boldsymbol{x_D}}) + \Delta^{cap,S}(\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) + \Delta^{cap,P}(\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) \\
&\quad + \Delta^{cap,W}(\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) + \Delta^{dem} \\
\Delta^{cap,S}(\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) &= \sum_{h\in L^S} \sum_{p\in P} cap^S_{h,p} \cdot (\hat{x^S_h} \cdot \upsilon^S_{h,p} \quad + \quad (1 - \hat{\phi^S_h}) \cdot \nu^S_{h,p} \\
&\quad + \quad \sum_{f\in F} (1 - u_f \cdot \hat{\psi^S_{h,p,f}}) \cdot \gamma^S_{h,p,f}) \\
\Delta^{cap,P}(\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) &= \sum_{i\in L^P} \sum_{b\in B} cap^P_{i,b} \cdot (\hat{x^P_i} \cdot \upsilon^P_{i,b} \quad + \quad (1 - \hat{\phi^P_i}) \cdot \nu^P_{i,b} \\
&\quad + \quad \sum_{f\in F} (1 - u_f \cdot \hat{\psi^P_{i,b,f}}) \cdot \gamma^P_{i,b,f}) \\
\Delta^{cap,W}(\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}) &= \sum_{j\in L^W} \sum_{p\in P} cap^W_{j,p} \cdot (\hat{x^W_j} \cdot \upsilon^W_{j,p} \quad + \quad (1 - \hat{\phi^W_j}) \cdot \nu^W_{j,p}) \\
\Delta^{dem} &= \sum_{k\in L^C} \sum_{p\in P} dem^C_{k,p} \cdot \delta^C_{k,p}
\end{aligned} \tag{5.15}$$

5.3.3. Single-level operator-attacker model: mOA-MIQP

mO-DUAL defined above will not be solved directly, but will be used to construct **mOA-MIQP**. By replacing the inner (minimization) problem of the bi-level **mOA** by its dual (maximization) problem **mO-DUAL**, one can create a single-level maximization problem **mOA-MIQP**. Note that the $\hat{\boldsymbol{x_A}}, \hat{\boldsymbol{x_D}}$ parameters in **mO-DUAL**, become decision variables $\boldsymbol{x_A}, \boldsymbol{x_D}$ in **mOA-MIQP**. This also means that **mOA-MIQP** is a quadratic, i.e. a non-linear, program (MIQP). Its quadratic terms may be linearized, but modern solvers can handle this type of MIQP relatively efficiently. See table 28 for an overview of the models presented in the paper.

Table 5.1: mO-DUAL dual variable definitions.

Constraints **mO**	Description	Dual variables **mO-DUAL**
4.2	Supplier capacity	$\upsilon^S_{h,p}, \quad \nu^S_{h,p}, \quad \gamma^S_{h,p,f}$
4.3	Producer capacity	$\upsilon^P_{i,b}, \quad \nu^P_{i,b}, \quad \gamma^P_{i,b,f}$
4.4	Warehouse capacity	$\upsilon^W_{j,p}, \quad \nu^W_{j,p}$
4.5	Supplier flow balance outgoing	$\alpha^S_{h,p}$
4.6	Warehouse flow balance	$\alpha^{W,in}_{j,p}, \quad \alpha^{W,out}_{j,p}$
4.7	Customer flow balance incoming	$\alpha^C_{k,p}$
4.8	Production BOM balance incoming flows	$\beta^{P,in}_{i,p}$
4.9	Production BOM balance outgoing flows	$\beta^{P,out}_{i,p}$
4.10	Demand and delivery balance	$\delta^C_{k,p}$
4.11	Transport load	$\mu_{i,j}$

Table 5.2: mO-DUAL dual constraints definitions.

Primal variables **mO**	Sets	Dual constraints **mO-DUAL**
$q^S_{h,p}$	$\forall h \in L^S, \quad \forall p \in P$	5.6
$q^P_{i,b}$	$\forall i \in L^P, \quad \forall b \in B$	5.7
$q^W_{j,p}$	$\forall j \in L^W, \quad \forall p \in P$	5.8
$q^C_{k,p}$	$\forall k \in L^C, \quad \forall p \in P$	5.9
$\bar{q}^C_{k,p}$	$\forall k \in L^C, \quad \forall p \in P$	5.10
$y_{i,j,p}$	$\forall (i,j) \in A, \quad \forall p \in P(i,j)$	5.11
$z_{i,j,m}$	$\forall (i,j) \in A, \quad \forall p \in M(i,j)$	5.12

mOA-MIQP $(\hat{\boldsymbol{x_D}}) \rightarrow \boldsymbol{x_\Omega}^*, \boldsymbol{x_A}^*$ is defined as follows:

[**mOA-MIQP**]

$$\max_{\boldsymbol{x_\Omega},\boldsymbol{x_A}} \quad \Delta^{oper,dual} \quad (\hat{\boldsymbol{x_D}}) \tag{5.15}$$

$$\text{s.t.} \quad \upsilon^S_{h,p} + \nu^S_{h,p} + \sum_{f\in F} \gamma^S_{h,p,f} - \alpha^S_{h,p} \quad \leq \quad c^{pr,S}_{h,p} \cdot \rho^c \qquad h \in L^S, \quad \forall p \in P \tag{5.6}$$

$$\upsilon^P_{i,b} + \nu^P_{i,b} + \sum_{f\in F} \gamma^P_{i,b,f} - \sum_{p\in P} (g^{in}_{b,p} \cdot b^{in}_{i,p} - g^{out}_{b,p} \cdot b^{out}_{i,p}) \quad \leq \quad c^{pr,P}_{i,b} \cdot \rho^c \qquad i \in L^P, \quad \forall b \in B \tag{5.7}$$

$$\upsilon^W_{j,p} + \nu^W_{j,p} - \alpha^{W,in}_{j,p} - \alpha^{W,out}_{j,p} \quad \leq \quad c^{pr,W}_{j,p} \cdot \rho^c \qquad j \in L^W, \quad \forall p \in P \tag{5.8}$$

$$\delta^C_{k,p} - \alpha^C_{k,p} \quad \leq \quad 0 \qquad k \in L^C, \quad \forall p \in P \tag{5.9}$$

$$\delta^C_{k,p} \quad \leq \quad c^{rv,C}_{k,p} \cdot \rho^r \qquad k \in L^C, \quad \forall p \in P \tag{5.10}$$

$$\alpha^S_{i,p} + \alpha^{W,in}_{j,p} + \alpha^{W,out}_{i,p} + \alpha^C_{j,p} + \beta^{in}_{i,p} + \beta^{out}_{i,p} + lc_p \cdot \mu_{i,j} \leq \quad 0 \qquad (i,j) \in A, \quad \forall p \in P(i,j) \tag{5.11}$$

$$-\, ls_m \cdot (1 - \tau_{i,j,m}) \cdot \mu_{i,j} \leq \quad c^{tr}_{i,j,m} \cdot \rho^c \qquad (i,j) \in A, \quad \forall m \in M(i,j) \tag{5.12}$$

$$\Gamma^{attacks} \quad \leq \quad bdg^{att} \tag{4.12}$$

$$\boldsymbol{\upsilon}, \quad \boldsymbol{\nu}, \quad \boldsymbol{\gamma}, \quad \boldsymbol{\beta}, \quad \boldsymbol{\mu} \leq \quad \mathbf{0} \tag{5.13}$$

$$\boldsymbol{\alpha}, \quad \boldsymbol{\delta} \quad \mathbb{R} \tag{5.14}$$

$$\boldsymbol{\phi}, \quad \boldsymbol{\psi}, \quad \boldsymbol{\tau} \in \quad \{0,1\} \tag{4.15}$$

5.4. Overall resolution approach

The general resolution approach is outlined in Algorithm **??**. It returns the optimal values for operational ($\boldsymbol{x_O}$), attacker ($\boldsymbol{x_A}$), and defender ($\boldsymbol{x_D}$) variables. The algorithm demonstrates that the full tri-level model (**mOAD**) is only required when both attacker (bdg^{ATT}) and defender (bdg^{DEF}) budgets are nonzero. If either budget is zero, only the relevant sub-model needs to be solved, providing a simple optimization. An overview of the different models presented in this section is given in table 28.

Algorithm 2 | General resolution approach

Input: $bdg^{ATT}, bdg^{DEF}, \hat{\boldsymbol{x_A}}$ (optional), $\hat{\boldsymbol{x_D}}$ (optional)
Output: $\boldsymbol{x_O}, \boldsymbol{x_A}, \boldsymbol{x_D}$

```
if bdg^ATT ≤ 0 then
    if bdg^DEF ≤ 0 then
        Solve mO(x̂_A, x̂_D) for x_O*  x_O ← x_O*
    else
        Solve mOD(x̂_A) for (x_O*, x_D*)  (x_O, x_D) ← (x_O*, x_D*)
else
    if bdg^DEF ≤ 0 then
        Solve mOA-MIQP(x̂_D) for (x_O,dual*, x_A*)  Solve mO(x̂_D, x_A*) for x_O*  (x_O, x_A) ← (x_O*, x_A*)
    else
        Solve mOAD for (x_O*, x_A*, x_D*) using algorithm 1  (x_O, x_A, x_D) ← (x_O*, x_A*, x_D*)
```

6 Results

This section presents the results of our study, divided into two main subsections. The first involves an exploration of the model's computational performance, and the second delves into a real-world case study: climate resilience of pharmaceutical company.

6.1. Computational performance

We examine the computational performance of **mOAD** by conducting multiple experiments with various system instances. These instances are defined by different combinations of system attributes to explore their impact on performance and identify the limits of tractability.

6.1.1. Instance definition

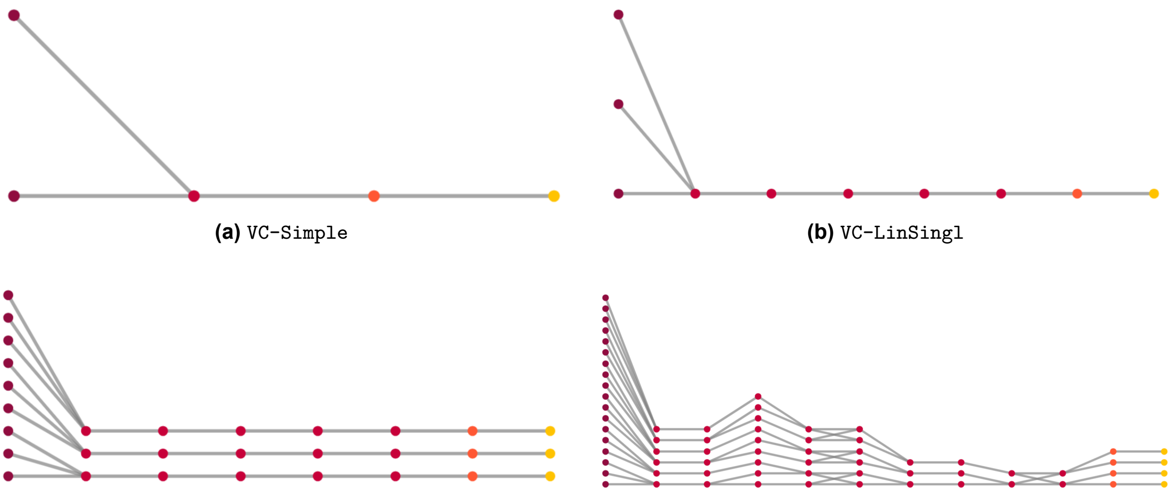
We define four main attributes according to which we define our supply chain run instances. The first attribute is the layout of the value chain (or production graph), which refers to the number of steps and interconnections within the production graph. The second attribute is the decision network size, which encompasses the number of customers and facilities, such as suppliers, producers, and warehouses, involved in model's system design and operational decisions. The third is the attack budget bdg^{ATT}, which are the varying amounts of resources used by the attacker component of the model to cause system disruptions. Lastly, the design / defense budget bdg^{DEF} involves varying levels of initial investment for system design and expansion. The systems for these experiments were generated randomly (random locations, commodity/production step/location names, demand, and supply capacities) but remained consistent across the runs.

Network size and complexity

The value chain of the supply chain is modeled through the production graph, consisting of `CONVERT` nodes for each bill of material/production step, and `SOURCE`, `STORE`, and `SINK` nodes for each commodity. Complexity arises not only from the number of steps but also from their interconnections and the various paths commodity flows can take. A previously noted, the production graph does not represent actual locations but abstract steps that can occur at any number of locations. This distinction between network complexity and size is important in our experiments. We define the four different value chain layouts, figure 6.1 depicts their layouts.

Value chain layout `VC-Simple` involves single production step with two input raw materials and one output final product, e.g. a simple food production process. `VC-LinSingl` represents a linear value chain with multiple production steps, two input raw materials, and one output final product, e.g. a basic manufacturing process. `VC-LinMulti` involves multiple parallel linear chains producing multiple final products, such as in a diversified manufacturing supply chain. `VC-Complex` is a complex chain with many input raw materials and production steps, and four output finished goods. This models a complex, industrial business such as an automotive manufacturer.

We vary the size of the decision network, which impacts both the number of possible system design configurations and operational settings/disruptions by the interdictor. The mathematical tri-level model does not distinguish between current and potential future locations. Three levels of decision network sizes are defined: `Small`, for a small, generally local supply chain with only a handful suppliers, producers, warehouses, and customers/demand points. `Medium` is an intermediate-sized decision network. `Large` has numerous (potential) facilities and demand points, i.e. a global-level supply chain.



(a) `VC-Simple` **(b)** `VC-LinSingl` **(c)** `VC-LinMulti` **(d)** `VC-Complex`

Figure 6.1: Production graphs of the different value chains.

Table 6.1: Instance definition - Value chains and decision network sizes

Value chain	# Production steps	# Commodities
`VC-Simple`	1	3
`VC-LinSingl`	5	8
`VC-LinMulti`	15	24
`VC-Complex`	43	74

Size	# Suppliers	# Producers	# Warehouses	# Customers
`Small`	10	5	5	10
`Medium`	30	15	15	50
`Large`	50	25	25	200

Attack and system design budgets

We conduct separate runs for different combinations of attack and defense/design budgets, representing resources for causing disruptions and for designing/expanding the system to be more resilient and efficient. These budgets are generally expressed in monetary units, with the exact values being less relevant than their relative impact. For disruptions, we first define four disruption levels $f \in F$: `Minor`, `Heavy`, `Major` and `Fatal`. These disruption levels correspond to varying (10, 20, 50 and 100%) reductions in process capacity at a location, and have increasing attack costs. The attack budgets determine how many simultaneous disruptions can be caused (see Table 6.2). Similarly, defense budgets determine the number of new facilities that can be established. By convention, existing facilities in the network have their initial costs set to zero. Introducing a new supplier incurs a relatively low initial cost, whereas establishing a new producer entails a significantly higher expense. The cost for a new warehouse falls somewhere in between. the relation between bdg^{DEF} and the cost of establishing a new facility is shown in table 6.2.

Table 6.2: Instance definition - Attack and design budgets

	Disruption level						New facilities		
bdg^{ATT}	Minor	Heavy	Major	Fatal	Link mode	bdg^{DEF}	# Suppliers	# Producers	# Warehouses
None	0	0	0	0	0	**None**	0	0	0
Mild	1	0	0	0	1	**Minimal**	1	0	0
Moderate	25	5	1	0	2	**Moderate**	10	0	1
Severe	50	12	2	0	4	**Extensive**	100	1	10
Catastrophic	200	50	8	2	6	**Global**	500	5	50

6.1.2. Results - Location-bound process disruptions

The following section presents the results and analysis of our computational experiments focusing on location-bound process disruptions, while neglecting link-based transport mode attacks. All computations were performed using the Gurobi solver with Python bindings on a core i5 processor. Tables 6.3 to 6.6 display the results. Each run was constrained by a time limit of 90 minutes (5400 seconds). We report the **solve time** in seconds and the remaining **gap** between the upper bound of the attacker sub-problem and the lower bound of the master problem, denoted as z^{UP} and z^{LO} respectively. The minimum gap ε was set to 10^{-5}. Instances with non-negligible gaps are highlighted in bold. Additionally, we report **Delivery**, a resilience metric indicating the percentage of total demand delivered, as a sanity check.

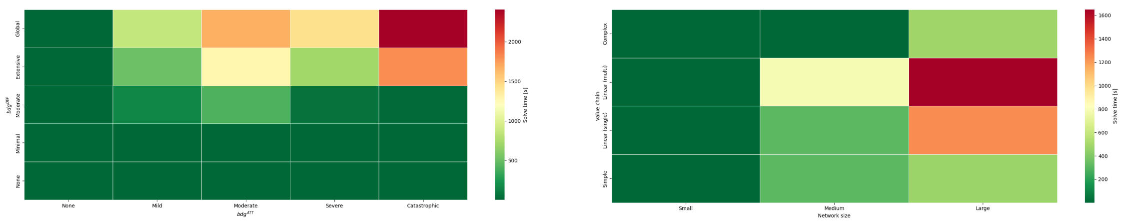


Figure 6.2: Performance grid of bdg^{ATT} and bdg^{DEF}, and network size and value chain complexity versus average solve time.

Regarding the computational performance of **mOAD**, we find that the decision network size significantly impacts solve time. This is predictable, as a larger decision network involves more system design decision variables ($\mathbf{x_D}$), which results in a larger **mOAD-MASTER**, the performance bottleneck of the algorithm. Interestingly, the complexity of the value chain and its production graph does not directly correlate with solve time. While more production steps introduce more operational setting decision variables ($\mathbf{x_A}$), this does not necessarily lead to longer solve times. Instead, complex value chains with more interconnections between steps are likely easier to disrupt due to the higher number of common processes involving large parts of the system. Consequently, such systems are more vulnerable, allowing the solver to find an optimal attack solution more quickly. The model performs worse when multiple disconnected graphs are involved, such as in the case of `VC-LinMulti`.

The size of bdg^{ATT} and bdg^{DEF}, relative to the available disruption and system design costs, also affects solve time. Generally, larger budgets result in longer solve times because problems with larger budgets are less constrained in $\mathbf{x_A}$ and $\mathbf{x_D}$, leading to a wider solution space. Conversely, lower budgets cause the model to discard solutions more quickly. There is also an interplay between bdg^{ATT} and bdg^{DEF} regarding solve time, as seen in figure 6.2. When both are large, the solution space grows non-linearly. Specifically, setting either budget to zero simplifies the tri-level **mOAD** into the more straightforward **mO**, **mOD**, and **mOA-MIQP** problems, which are solvable almost instantly, typically in less than a second. Our findings indicate that **mOAD**'s solve time is most sensitive to decision network size, compared to value chain complexity. Overall, the computational experiments demonstrate that the model is solvable within a reasonable time frame for a wide range of realistically sized cases.

6.1.3. Deterministic attacker sub-problem versus simulation

An interesting experiment to further demonstrate the model's computational capability is to compare it to a simulation approach. The attacker sub-problem **mOA-MIQP** returns the optimal (i.e., worst)

Table 6.3: Computational results for `VC-Simple`

Network size		Small					Medium					Large				
	bdg^{DEF}	None	Minimal	Moderate	Extensive	Global	None	Minimal	Moderate	Extensive	Global	None	Minimal	Moderate	Extensive	Global
bdg^{ATT}	Results															
None	**Solve time [s]**	0.00	0.01	0.01	0.02	0.01	0.01	0.01	0.10	0.42	0.12	0.04	0.06	0.39	2.06	2.62
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.03	0.18	0.24	0.10	0.08	0.13	1.12	3.78	5.59	0.60	0.53	2.20	71.39	3.88	4.43
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00
Moderate	**Solve time [s]**	0.03	0.15	0.30	0.12	0.12	0.12	0.59	41.98	261.42	0.81	0.47	2.25	392.79	3.90	4.44
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Severe	**Solve time [s]**	0.04	0.14	0.26	0.19	0.12	0.12	0.63	42.23	263.82	0.73	0.46	2.25	382.06	4.05	4.70
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Catastrophic	**Solve time [s]**	0.03	0.24	0.24	2.49	5.34	0.13	1.63	1.49	**5400.00**	1787.82	0.53	5.34	5.88	**5400.00**	5400.00
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**173.64**	0.00	0.00	0.00	0.00	**174.23**	0.58
	Delivered [%]	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	100.00	0.00	0.00	0.00	100.00	100.00

Table 6.4: Computational results for `VC-LinSingl`

Network size		Small					Medium					Large				
	bdg^{DEF}	None	Minimal	Moderate	Extensive	Global	None	Minimal	Moderate	Extensive	Global	None	Minimal	Moderate	Extensive	Global
bdg^{ATT}	Results															
None	**Solve time [s]**	0.01	0.01	0.02	0.02	0.02	0.01	0.02	0.22	0.32	0.14	0.06	0.07	0.29	1.43	1.77
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.06	0.53	0.77	0.57	0.69	0.21	1.11	4.41	12.80	9.48	0.73	2.96	1023.97	586.64	387.14
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00
Moderate	**Solve time [s]**	0.06	0.41	0.62	2.60	0.59	0.29	1.15	5.86	84.93	21.81	0.61	3.00	57.81	**5400.00**	5190.04
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**50.96**	0.21
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Severe	**Solve time [s]**	0.05	0.24	0.55	1.38	0.81	0.20	0.88	5.66	11.84	11.38	1.62	4.60	107.02	2132.22	5283.48
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Catastrophic	**Solve time [s]**	0.08	0.28	0.73	3.70	4.35	0.27	1.26	6.04	2299.32	**5400.00**	0.68	3.10	7.71	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**102.87**	0.00	0.00	0.00	**101.71**	**103.23**
	Delivered [%]	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	100.00

disruption set, $\mathbf{x}^*_{\mathbf{A}}$, given a system design, $\mathbf{x_D}$, using deterministic optimization. We can define **mOA-SIMUL**, another version of the attacker sub-problem that employs Monte Carlo simulation to identify suitable attacks given an attack budget bdg^{ATT}. **mOA-SIMUL** takes a system design, $\mathbf{x_D}$, as input and generates a set of N randomly generated valid operational setting vectors, $\mathbf{x^k_A}$. It then measures their effect on system performance by solving **mO**($\hat{\mathbf{x_A}}^{\mathbf{k}}$, $\hat{\mathbf{x_D}}$).

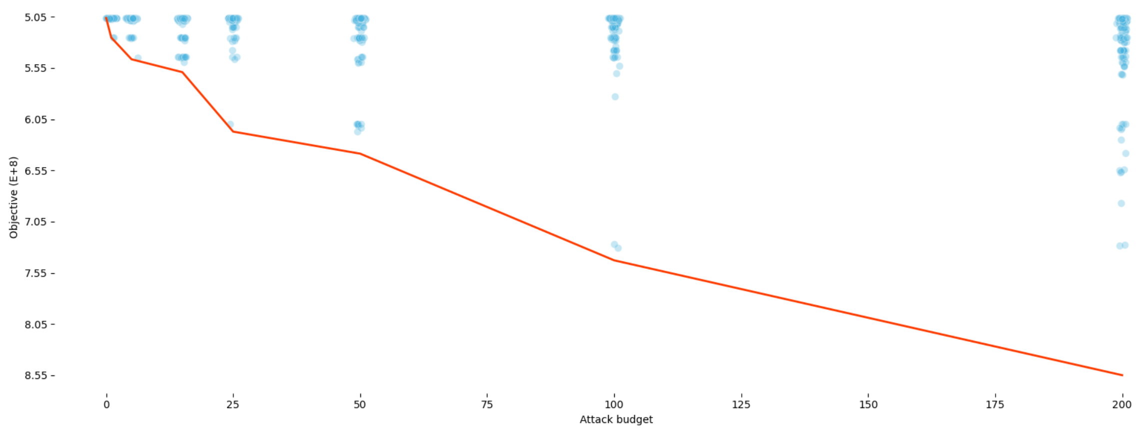


Figure 6.3: Objective value of optimal **mOA-MIQP** disruption $\mathbf{x}^*_{\mathbf{A}}$ (orange line) versus 200 random attacks, per budget (blue pointcloud).

Figure 6.3 presents the results of running the **mOA-MIQP** versus **mOA-SIMUL** experiment on a random `MEDIUM`-sized supply chain for various attack budgets, with **mOA-SIMUL** using 200 random attacks per bdg^{ATT}. For most budgets, **mOA-MIQP** is solved in under a second. Running numerous instances of **mO** is significantly more resource-intensive, as evidenced by Figure 6.3, where **mOA-SIMUL** struggles to find attacks that are anywhere near optimal. The higher the bdg^{ATT}, and thus the larger the solution space, the more simulations are required to find optimal attacks. This highlights why a deterministic

Table 6.5: Computational results for `VC-LinMulti`

	Network size	`Small`					`Medium`					`Large`				
	bdg^{DEF}	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**
bdg^{ATT}	*Results*															
None	**Solve time [s]**	0.01	0.01	0.01	0.01	0.01	0.02	0.05	0.11	0.17	1.22	0.09	0.12	0.63	2.44	2.73
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.10	0.47	0.49	0.42	0.50	0.32	1.42	6.73	62.16	2885.99	2.37	10.10	594.24	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**10.73**	**10.98**
	Delivered [%]	96.67	96.67	96.67	96.67	96.67	96.67	96.67	96.67	96.67	100.00	96.67	96.67	96.67	100.00	100.00
Moderate	**Solve time [s]**	0.22	0.49	0.47	0.75	0.49	0.42	1.59	65.08	3977.50	5357.82	2.24	16.78	4290.81	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**15.08**	**66.13**
	Delivered [%]	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	83.33	100.00	80.00	80.00	86.67	93.33	100.00
Severe	**Solve time [s]**	0.22	0.58	0.61	0.80	0.54	0.39	2.65	11.62	568.13	1077.06	1.24	15.65	92.26	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**35.42**	**108.72**
	Delivered [%]	66.67	66.67	66.67	66.67	66.67	66.67	66.67	66.67	66.67	100.00	66.67	66.67	66.67	100.00	100.00
Catastrophic	**Solve time [s]**	0.31	0.93	1.47	2.39	1.42	0.54	5.25	5.13	195.72	**5400.00**	1.36	14.35	161.47	2857.83	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**35.29**	0.00	0.00	0.00	0.00	**221.10**
	Delivered [%]	33.33	33.33	33.33	33.33	33.33	33.33	33.33	33.33	33.33	100.00	33.33	33.33	33.33	33.33	100.00

Table 6.6: Computational results for `VC-Complex`

	Network size	`Small`					`Medium`					`Large`				
	bdg^{DEF}	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**
bdg^{ATT}	*Results*															
None	**Solve time [s]**	0.02	0.01	0.03	0.02	0.03	0.09	0.26	0.10	0.08	0.15	0.17	0.17	0.34	0.34	0.48
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.55	2.27	2.04	2.48	2.70	1.24	4.10	4.30	3.62	3.58	2.06	9.39	24.80	77.03	1672.71
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00
Moderate	**Solve time [s]**	0.67	2.28	2.36	2.44	2.32	0.99	4.39	7.66	4.41	4.32	3.35	10.88	24.99	200.51	4175.57
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Severe	**Solve time [s]**	0.44	2.16	2.19	1.96	2.05	0.88	6.42	6.60	3.62	3.59	2.50	11.37	40.83	166.07	5400.00
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Catastrophic	**Solve time [s]**	0.55	3.82	3.16	3.19	3.22	0.80	3.51	3.63	3.60	3.54	2.15	9.74	17.85	33.09	92.05
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

approach for identifying critical aspects of a supply chain network is better suited than a simulation approach.

6.2. A case study: climate resilience of a global pharmaceutical company

This case study revolves around an international pharmaceutical company. While the data has been anonymized and slightly randomized, the production steps remain realistic, and the facility locations are approximate. Expansion plans are invented to provide a practical application of our resilient supply chain model.

6.2.1. Case study description

The pharmaceutical company produces three drugs: `ProductA`, `ProductB`, and `ProductC`. These products are primarily sold in the US and Europe, with additional demand in Latin America, South Africa, and the Asia-Pacific region. The production process includes drug substance production, vial filling, and packaging. Each drug substance is produced from raw materials supplied by various biological suppliers. `ProductA` and `ProductB` are filled into vials with varying contents: 10, 20, and 30 mg for `ProductA`, and 5, 10, 15, 20, and 25 mg for `ProductB`. The filled vials are then packed into units containing 1, 2, 5, 6, or 10 vials, which are subsequently distributed to customers worldwide.

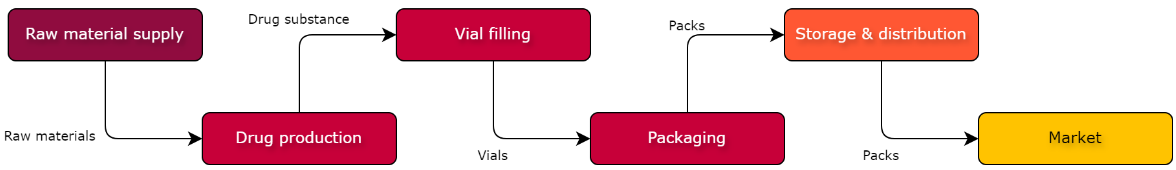


Figure 6.4: Diagram of production steps.

Figure 6.4 presents a simplified version of the company's value chain. This simplified representation is

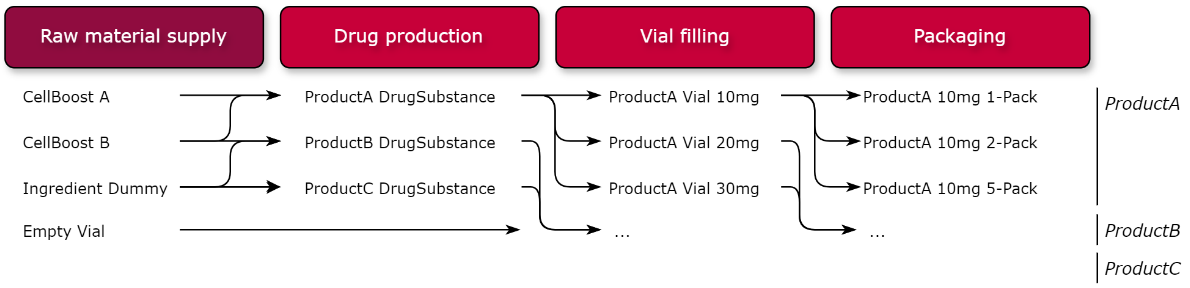


Figure 6.5: Commodities.

expanded to depict the various product combinations in Figure 6.5, illustrating some of the commodities involved, including raw materials, intermediate, and final goods. The production graph for this system, shown in Figure 6.6, was constructed as described in section 4.1. It displays, from left to right, the array of `SOURCE` nodes for different raw materials, `CONVERT` nodes for substance production, filling, and packaging, and `STORE` and `SINK` nodes for the final products. This expanded value chain reveals the inherent complexity of the production process.

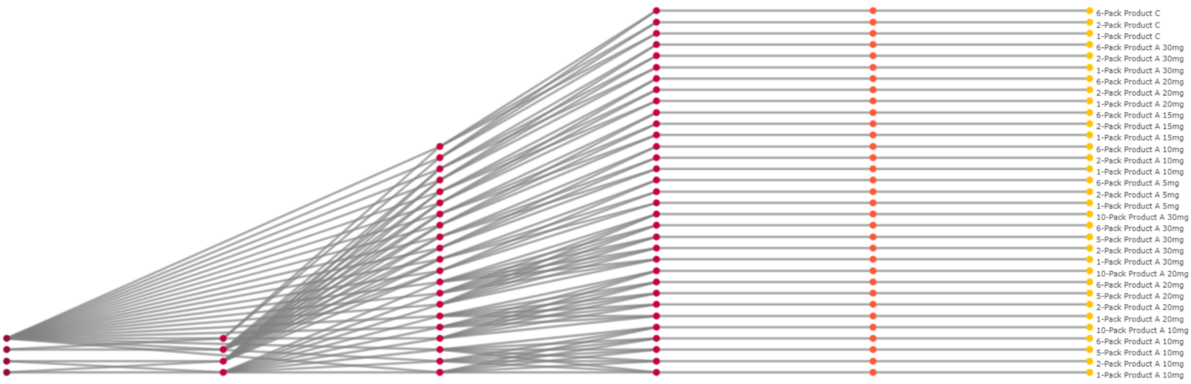


Figure 6.6: Production graph of the pharmaceutical company.

Current network

The company's current supply chain comprises suppliers, production sites, warehouses, and demand cities. There are four production facilities: `MU-UnitedStates`, `MU-Belgium`, `MU-Italy`, and `MU-Ireland`. These facilities differ in the production steps they handle and the products they produce. Each facility manages various production resources, including substance production units, filling units, and packaging units. For instance, the `MU-Belgium` facility has multiple filling lines, or machines: TL, Syntegon, and Bosch. Packs of 5 and 10 vials are produced exclusively in the US, while packs of 1 and 6 vials are produced in Europe, and packs of 2 vials are produced in both regions. Figure 6.7 details the production resources at each facility.

The distribution network includes approximately 30 warehouses worldwide, connected to 189 demand cities. Each demand point may require different sub-products. The transportation modes defined on links between suppliers, producers, and warehouses are truck, rail, and sea (where logical), while connection to customers is primarily through LTLs.

Network expansion

The business aims to expand the current system by incorporating potential new locations to handle increased demand, particularly from Asia. These new locations are also included in the Supply Chain Decision Network. Potential new production sites have been identified in Germany, India, Indonesia, Egypt, and Brazil, each with varying production resources and capacities. Additionally, ten new raw material suppliers and 30 potential warehouse locations have been identified, each with different investment costs. Figure 6.8 shows the current locations in color and potential locations in grey.

Figure 6.7: Current production resources.

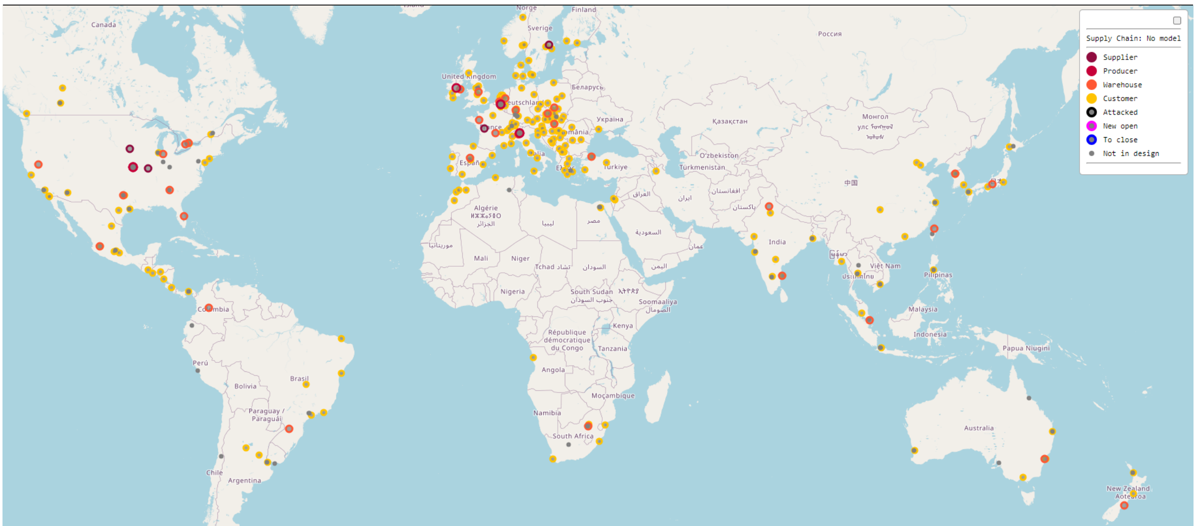


Figure 6.8: Location map of the case study.

6.2.2. Determining attack costs: A climate resilience approach

Our mathematical model introduces the concepts of attacker budget bdg^{ATT} and attacker costs c^{ATT}, representing the resources consumed by the interdictor in their optimisation of most disruptive operational setting. A deliberate design choice of the model is that users are responsible for defining these attack costs. In this case study, we will however provide an example based on climate hazards and geopolitical risks. To investigate the impact of resilience to climate risks, we associate attack costs with the risk of being impacted by various hazards. Each location i is assigned a risk score $\theta_{i,r}$ between 0 and 1 for each hazard $r \in R$. A risk-score of 0 indicates that the location should not realistically be affected by hazard r before 2050 while a score of 1 indicates a very high likelihood of being affected by the hazard before 2050. The attack cost $c_{i,f}^{ATT}$ for disrupting facility i by disruption level f is defined by Equation 6.1. Disruption levels are defined in table 6.7 and range from minimal impact (-10% capacity) to total disruption.

$$c_{i,f}^{ATT} = k_f \quad (1 - \sum_{r \in R} K_{r,f} \cdot \theta_{i,r}) \qquad (6.1)$$

In this equation, $\theta_{i,r}$ represents the susceptibility of facility i to hazard r. $K_{r,f}$ relates hazard r to disruption level f, with $0 \leq K_{r,f} \leq 0.05$. As such, we can model the different impacts different hazards may have. An earthquake is more likely to cause a `FATAL` disruption while heat may cause `MINOR` to `HEAVY` ones. k_f is the level bias, a constant specific to the disruption level. We can assume for example that a `MINOR` disruption is more likely to occur than a `FATAL` one. Figure 6.9 shows the risk scores for the drought hazard, compiled using the fema$_n rinationalriskindexdataandinternalclimateriskmodelsfromclimate_rhdhv.T$

Table 6.7: Climate hazard and disruption level definitions

Disruption level f	Impact u_f [%]	Level bias k_f	Risks $K_{r,f}$	Heat	Cold	Rain	Snow	TropicalStorm	Wildfire	AirQuality	Flood	Drought	Earthquake	HumanConflict
MINOR	10	1		0.020	0.020	0.020	0.020	0.005	0.005	0.010	0.005	0.020	0.005	0.005
HEAVY	20	4		0.020	0.020	0.020	0.020	0.010	0.010	0.005	0.010	0.020	0.010	0.010
MAJOR	50	25		0.010	0.010	0.010	0.010	0.050	0.050	0.005	0.050	0.010	0.050	0.050
FATAL	100	100		0.005	0.005	0.005	0.005	0.050	0.050	0.001	0.050	0.010	0.050	0.050

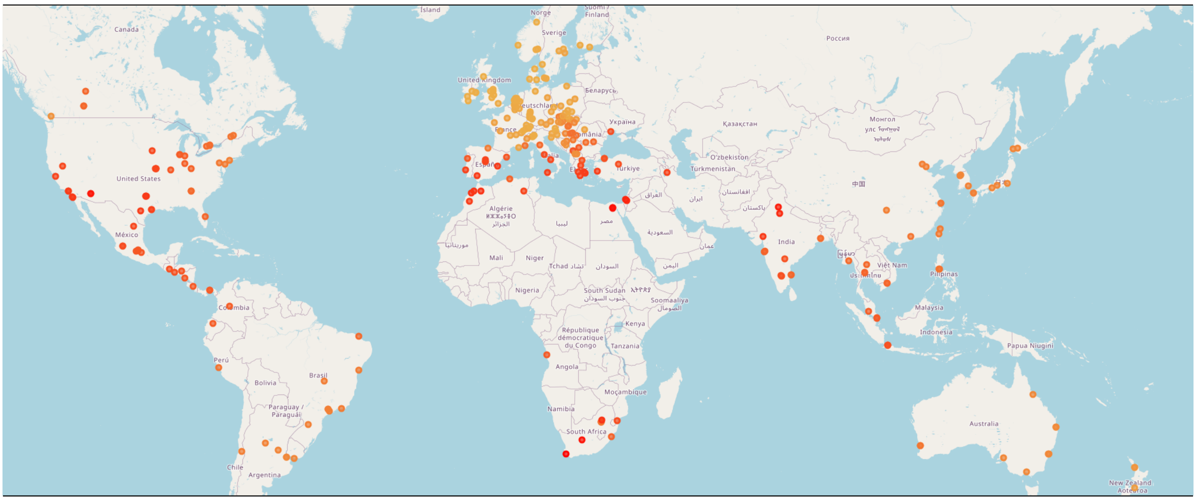


Figure 6.9: Map showing the climate risk factors for "Drought" for the relevant case study locations.

6.2.3. Use case analyses

We will now explore how the model can be utilized for resilient supply chain design and operations. We identify four main use cases for which this model is particularly relevant:

UC1 Operating an efficient supply chain.
UC2 Identifying vulnerable parts of a supply chain.
UC3 Designing / Expanding an efficient supply chain.
UC4 Designing / Expanding an efficient supply chain assuming disruptions may occur.

These use cases are common in supply chain management. **UC1** and **UC3** involve standard supply chain operations and design. While these aren't the primary focus of our model, they can still be effectively addressed within the same framework. **UC2** and **UC4**, however, emphasize resilience, with **UC4** allowing us to explore the trade-offs between efficiency and resilience in supply chain design. The different use cases can be addressed using **mOAD** by defining different values for bdg^{ATT} and bdg^{DEF}. **UC1** involves setting both bdg^{ATT} and bdg^{DEF} to 0. Solving **mOAD** with these budgets is equivalent to solving **mO**. **UC2** involves setting $bdg^{ATT} > 0$ and bdg^{DEF} to 0, equivalent to solving **mOA-MIQP**. **UC3** involves setting bdg^{ATT} to 0 and $bdg^{DEF} > 0$, which boils down to solving **mOD**. Finally, **UC4** involves setting both bdg^{ATT} and $bdg^{DEF} > 0$. Here, we solve the iterative tri-level **mOAD**.

UC1 Operating an efficient supply chain

This use case is illustrated in Figures 6.10 and 6.11, which show the result of **mO**: optimal operations of the current supply chain without disruptions. **mO** essentially solves a multi-commodity flow problem, where commodities flow from suppliers to demand cities, being transformed at production facilities. In Figure 6.10, facilities are represented as nodes, and commodity flows are shown as links between nodes, with thickness proportional to flow volume. For producers, the facilities are divided into main production processes (drug substance, filling, packaging) to better illustrate intra-facility flows. Each figure displays the values of the objective function terms: operational costs Γ^{cost} and penalties Γ^{penal}, weighted by ρ^c and ρ^r. In the base case, not all demand can be fulfilled yet.

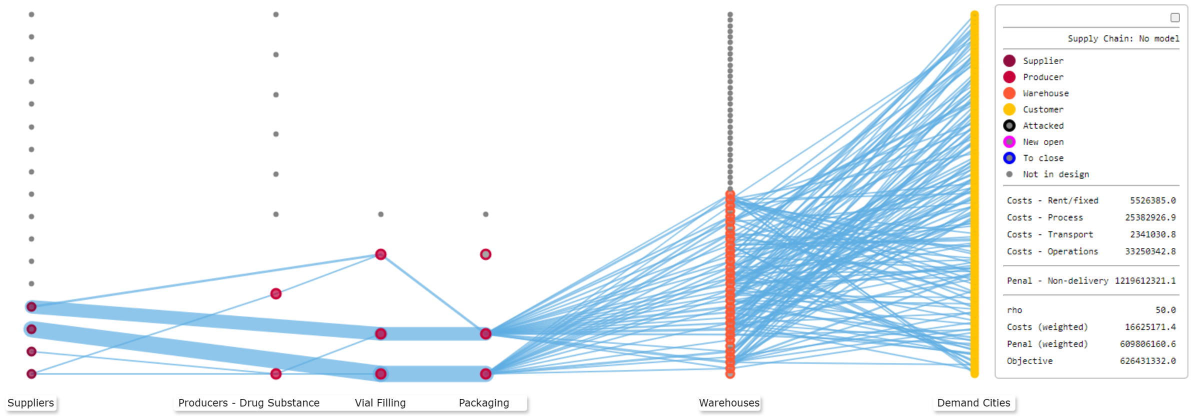


Figure 6.10: Flow diagram, $bdg^{ATT} = 0$, $bdg^{DEF} = 0$.

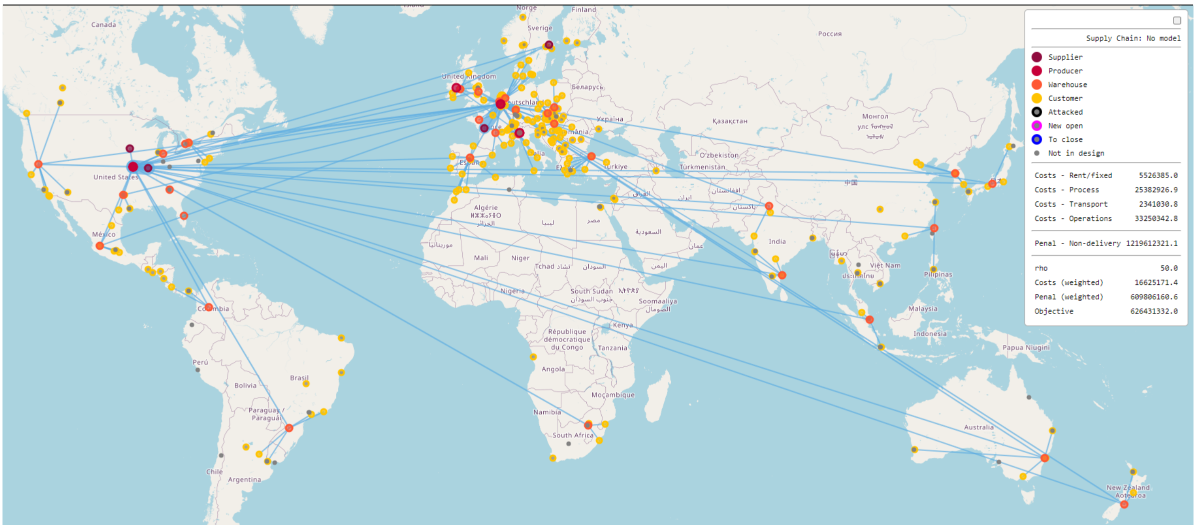


Figure 6.11: Map, $bdg^{ATT} = 0$, $bdg^{DEF} = 0$.

UC2 Identifying vulnerable parts of a supply chain

This use case is demonstrated in Figures 6.12 and 6.13, showing the impact of the worst-case combination of simultaneous disruptions with $bdg^{ATT} = 1$ (minor disruption) and $bdg^{ATT} = 100$ (major disruption or multiple smaller disruptions), respectively. The effect of the $bdg^{ATT} = 1$ scenario is relatively negligible, while the $bdg^{ATT} = 100$ scenario causes almost total system disablement. Attacked locations are marked in black. Table 6.8 lists the disrupted facilities at various disruption levels for different bdg^{ATT} values, with $bdg^{DEF} = 0$. It is noteworthy that for different attack budgets, the critical facilities identified change considerably, indicating that supply chain vulnerabilities are often due to a combination of multiple weak links rather than a single weak point.

Table 6.8: Disrupted facilities, by bdg^{ATT}, and with $bdg^{DEF} = 0$.

Facility	bdg^{ATT}	**0**	**1**	**5**	**25**	**50**	**100**	**200**	**300**
MU Belgium PACKAGING			MINOR	MINOR	HEAVY	MAJOR			FATAL
MU UnitedStates DRUGPROD				MINOR	MINOR	HEAVY	FATAL	FATAL	
MU UnitedStates PACKAGING				HEAVY	MAJOR	MAJOR			FATAL
SU CellBoost US					MINOR	MINOR			
MU Ireland DRUGPROD					MINOR	HEAVY	MAJOR	FATAL	
WH Germany GRIESHEIM						MINOR	MINOR		
WH NETHERLANDS Nijmegen							MINOR		
MU Italy PACKAGING									FATAL

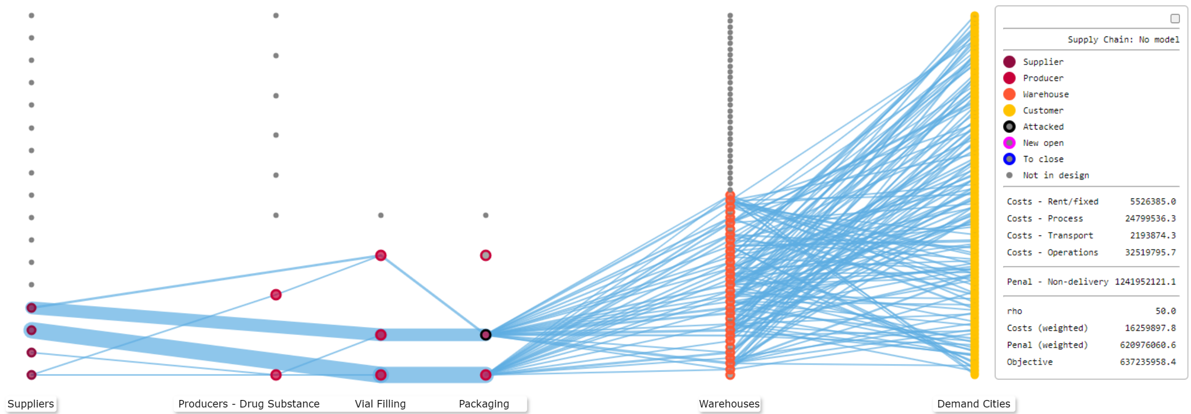


Figure 6.12: Flow diagram, $bdg^{ATT} = 1$, $bdg^{DEF} = 0$.

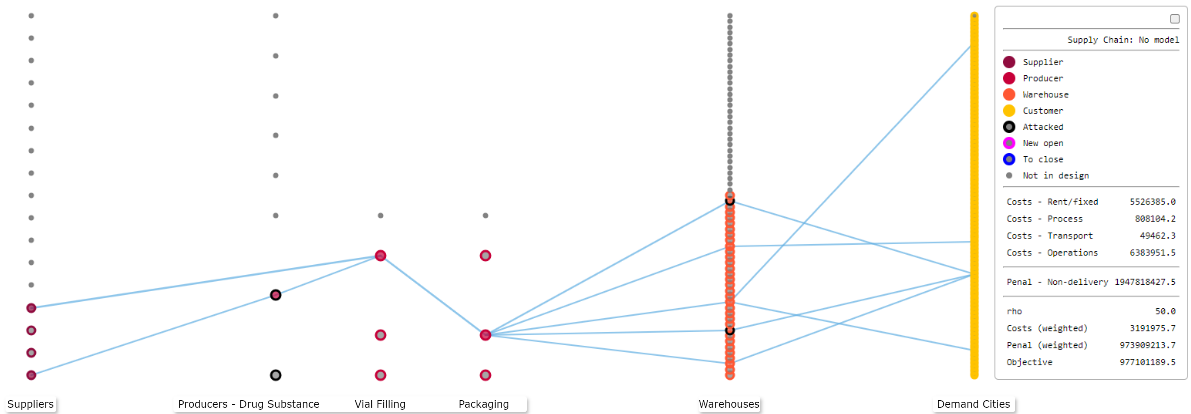


Figure 6.13: Flow diagram, $bdg^{ATT} = 100$, $bdg^{DEF} = 0$.

UC3 - Designing / Expanding an efficient supply chain.

Strategic decisions are involved for the first time in this use case, addressing system design decisions. By solving **mOD** using an initial/expansion investment budget, the model outputs the optimal system design for best system performance. Given a lower budget, the model may only establish a new supplier to improve operational performance, as shown in Figure 6.14. With a much higher budget, the model decides to open various new facilities, such as new producers and new warehouses, to lower operational transportation costs, as in Figure 6.15. The model makes no distinction between new and current locations, with current locations being distinguished solely by their nil initial cost. The model may also decide to stop using currently open locations if their advantages do not outweigh their fixed costs, such as rent and staff. On the flow diagrams, new locations in the system design are marked in magenta, while locations that were open but not included anymore are marked in blue. Table 6.9 shows the new locations established for various bdg^{DEF} and, similarly to the previous attack table, whether a certain location is included or not changes considerably with each different budget.

Table 6.9: Newly included facilities in the design, by bdg^{DEF}, and with $bdg^{ATT} = 0$..

New facility	0	10 000	20 000	5 000 000	10 000 000	20 000 000	50 000 000	100 000 000	200 000 000
SU CellBoost US TEXAS		✓	✓					✓	
SU CellBoost United Kingdom			✓					✓	✓
WH USA LEWISBERRY, PENSILVANIA				✓	✓	✓		✓	✓
WH Australia PERTH					✓	✓		✓	
WH Canada CALGARY						✓			✓
WH Germany KARLSRUHE						✓			
MU Brazil							✓	✓	✓
MU Germany PACKAGING								✓	✓
WH Brazil VINHEDO								✓	✓
SU CellBoost Singapore									✓
MU India									✓
MU Egypt									✓

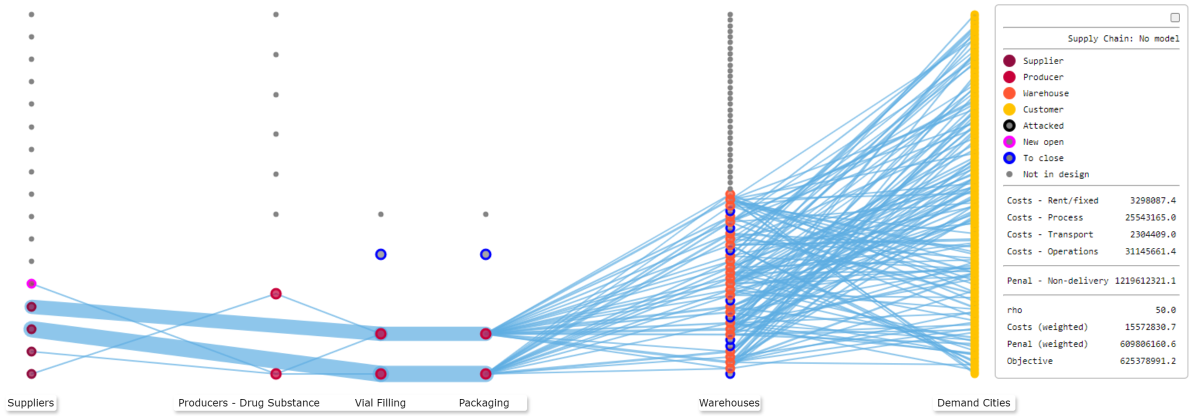


Figure 6.14: Flow diagram, $bdg^{ATT} = 0$, $bdg^{DEF} = 10000$.

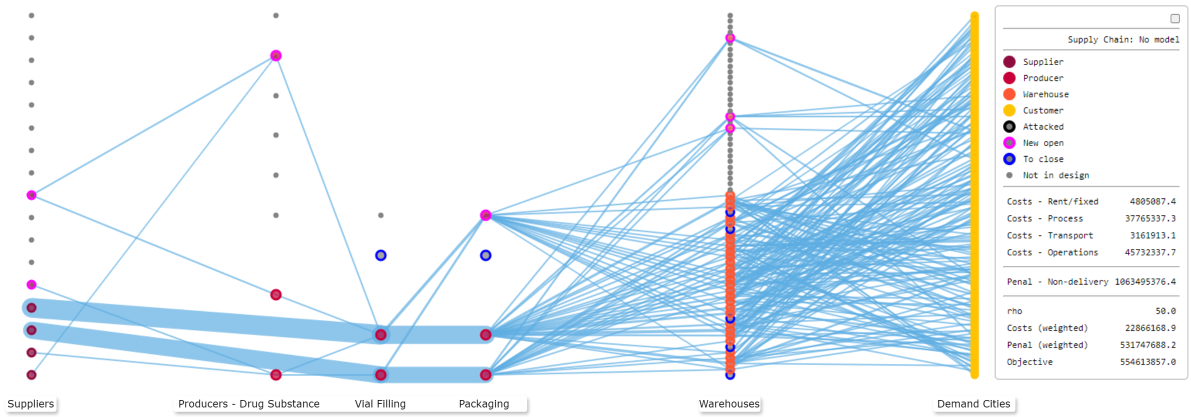


Figure 6.15: Flow diagram, $bdg^{ATT} = 0$, $bdg^{DEF} = 100000000$.

UC4 - Expanding an efficient supply chain assuming disruptions may occur

In this use case, we solve the full tri-level **mOAD**, considering operational settings and potential disruptions. Figures 6.17 and 6.18 show the results of optimal system designs for maximum system performance, anticipating worst-case disruptions. This trade-off between efficiency and resiliency is crucial in strategic system design. By running multiple scenarios with various combinations of bdg^{ATT} and bdg^{DEF}, we visualize the resilience of the system using resilience grids. These grids plot the budgets versus system performance, objective function value, or raw delivery satisfaction, as shown in

Figure 6.2. Cells in dark purple are instances that couldn't be solved with a optimality gap lower than 1% within the cut-off time of 90 minutes. Overall, these various runs highlight the model's capability to handle multiple types of strategies in response to disruptions. It can handle strategic decisions, such as multi-sourcing and capacity increases through the establishment of new facilities, as well as contingent decisions, like rerouting commodity flows during disruptions. This adaptability leads to more resilient supply chains capable of maintaining efficiency under varying operational settings.

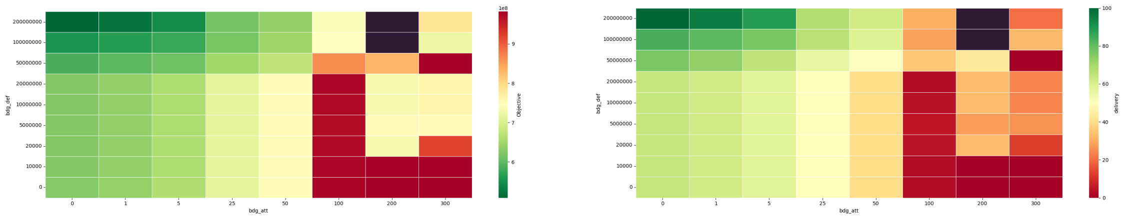


Figure 6.16: Resilience grids of bdg^{ATT} and bdg^{DEF} versus the objective value and the demand delivered [%].

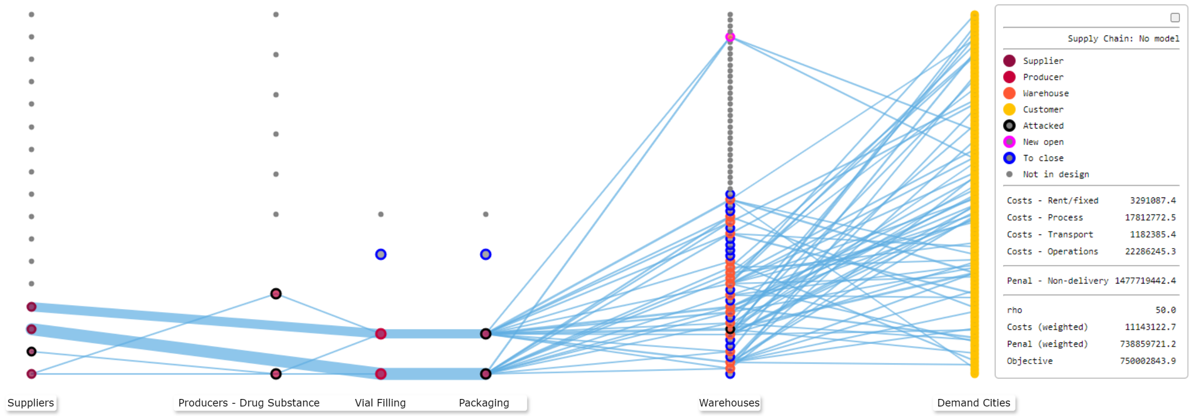


Figure 6.17: Flow diagram, $bdg^{ATT} = 50$, $bdg^{DEF} = 5000000$.

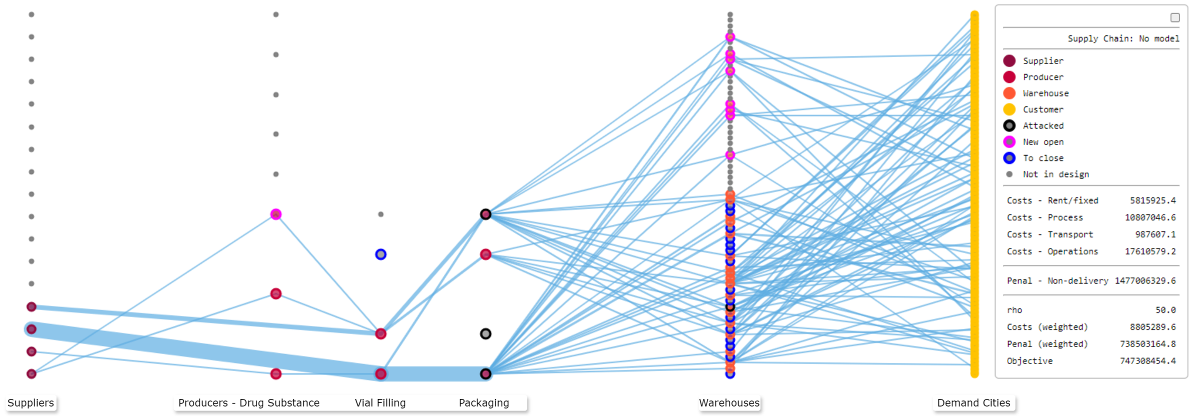


Figure 6.18: Flow diagram, $bdg^{ATT} = 100$, $bdg^{DEF} = 100000000$.

6.2.4. Adjusting system performance weights

To enhance flexibility, the proposed system performance function, which serves as the objective function of **mOAD**, includes weights ρ^c and ρ^r for the two main cost terms: operational costs Γ^{costs} and penalty costs Γ^{penal} (where $\rho^c + \rho^r = 1$). This allows model users to determine the emphasis placed on optimizing system resilience versus operational costs. Up to this point, every model run in this paper has used $\rho^c = \rho^r = 0.5$. We will now perform several runs varying ρ^r between 0 and 1, focusing on the case where $bdg_{ATT} = 50$ and $bdg_{DEF} = 100000000$. As anticipated, solutions with a low value of ρ^r will prioritize reducing operational costs over fulfilling demand, resulting in a higher tolerance for penalties. When $\rho^r = 0\%$, no demand is met, whereas when $\rho^r = 100\%$, all demand is fulfilled regardless of the cost. The optimal balance lies somewhere in between these extremes, and it is up to the user to determine a suitable middle ground for their specific needs. Of course, the model has been set up with this particular formulation of the performance cost function, but it can be adapted for other definitions of system performance.

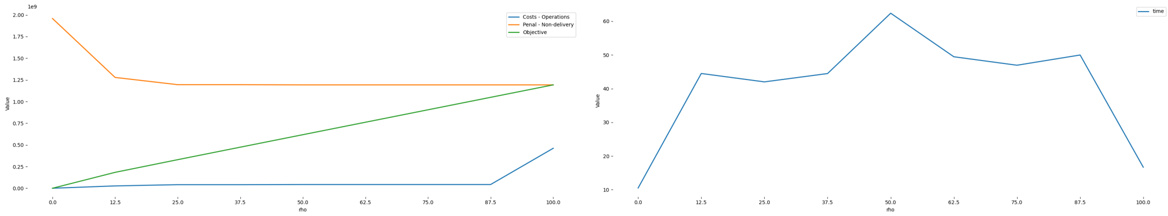


Figure 6.19: Performance function term values and solve time for different values of ρ^r.

7
Potential model extensions

7.1. Trans-shipments and multi-modal trips

When incidents like the Ever Given getting stuck in the Suez Canal or a ship colliding with a bridge in the Port of Baltimore occur, entire shipping lanes can grind to a halt, disrupting all supply chains reliant upon them. We can currently simulate such disruptions using the link attack variables $\tau_{i,j,m}$ within the operational setting vector $\mathbf{x_A}$. However, a more practical approach involves modeling these bottlenecks through transshipment nodes, which may model ports and canals.

In our production graph, we introduce a new category of `STORE` nodes specifically designed for this purpose. These transshipment nodes can be targeted by the attacker model like any other facility, thereby disrupting all flows passing through them in a single event. This approach enables to identify crucial and vulnerable nodes in the network, and also the modeling of truly multi-modal trips, where journeys involve multiple modes of transportation (e.g., sea and truck).

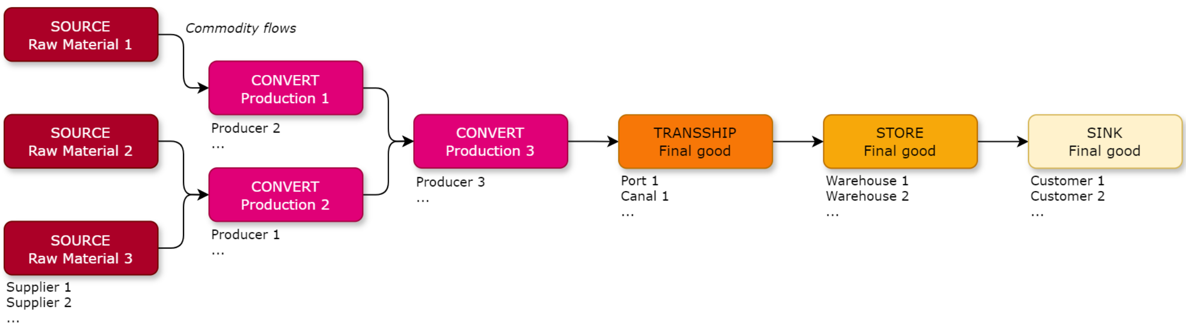


Figure 7.1: Example Production Graph with additional Transshipment node to model ports, canals, etc.

7.2. Going tactical: incorporating a temporal component

Up until now, our focus has predominantly been on the strategic level of supply chain resilience, analyzing the spatial dynamics of supplier networks, production facilities, and distribution channels. The model's decisions involved regard system design and aggregate network flows over a single, "longer" time-period. and does not involve a temporal dimension. However, to truly understand and enhance resilience, we must incorporate a temporal dimension into our analysis. A significant improvement to our study of resilience is the incorporation of a time component. Supplier orders, production processes, and transportation of goods all take time, and some commodities may have expiration dates or late-delivery penalties that affect the system's performance. Transforming our purely spatial supply chain conceptual model into a spatial-temporal one and adapting the mathematical model accordingly is an important topic for further research.

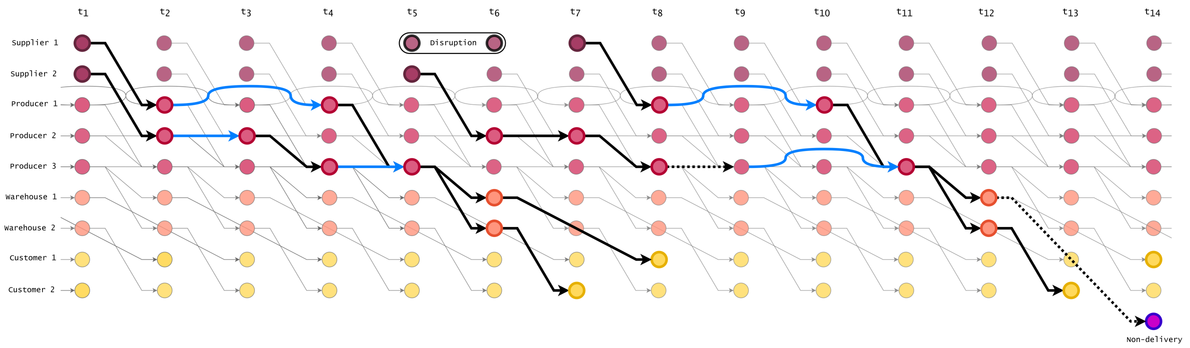


Figure 7.2: Incorporating a temporal component: a time-space diagram as conceptual model.

In Figure 7.2, we propose a time-space representation of the multi-commodity flow supply chain model as a starting point. This model includes different locations and the links between them, and also multiple time-steps. Links connect two time-space nodes and can represent commodity transport arcs between different locations (in black) and process arcs within a location (in blue), which may span multiple time-steps. Process arcs could represent production processes as shown, or the supply or storage of commodities that require some minimal process durations, and these arcs may have some storage capacity. The mathematical formulation should ensure that these minimal durations are respected. Figure 7.2 illustrates two examples of commodity flows within the time-space decision network. The left side presents a solution under normal conditions, while the right side depicts a solution in a disrupted scenario. Disruptions will have a duration, adding a temporal impact on operational responses in the form of delays and expired deliveries (dotted arcs). The disrupted scenario shows how Supplier 1 experiences a disruption between time-steps t_5 and t_6, leading to a supply delay. This delay propagates to Producer 3 at time-step t_8, ultimately causing the delivery of final goods to customers to be delayed. Consequently, some deliveries expire and are redirected to a dummy non-delivery node at t_{14}. Penalizing late or expired deliveries will surely require non-continuous variables in the operator problem, however. Note that item inventory and supplier order decisions per time-step can also be modeled.

8 Conclusion

This study adapts the Operator-Attacker-Defender (OAD) model for resilient supply chain design and operation. The tri-level optimization model addresses disruptions such as supplier failures, production shortfalls, and disabled transportation links through capacity reduction. It enables contingent rerouting and strategic design of supply chains for increased capacity and multi-sourcing. The model is decomposed into an attacker sub-problem and a master problem, solved iteratively.

The model was evaluated using a multitude of computational experiments and a real-life case study. The computational performance analysis reveals that decision network size significantly impacts solve time, while value chain complexity has a more nuanced effect. Larger decision networks involve more system design variables, leading to longer solve times due to the increased complexity of the master problem. Conversely, more complex value chains do not necessarily result in longer solve times; instead, they may be easier to disrupt, allowing the solver to find optimal solutions more quickly. The interplay between attack and defense budgets also influences solve time, with larger budgets generally expanding the solution space and increasing computational effort. The comparison between deterministic optimization and simulation approaches underscores the efficiency of the deterministic method in identifying optimal disruptions. The deterministic attacker sub-problem outperforms the simulation-based approach, particularly as the attack budget increases, highlighting the superiority of the deterministic framework for critical supply chain analysis. The real-world case study further validates the model's practicality. By analyzing an international pharmaceutical company's supply chain, we demonstrate how the model can identify vulnerable parts of the supply chain and suggest optimal expansions to enhance resilience. The case study showcases the model's ability to handle complex, real-world data and provide actionable insights for supply chain managers.

Based on our findings, we believe the OAD model is a valuable approach for supply chain managers to design and operate more resilient supply chains. This approach allows for a comprehensive assessment of vulnerabilities and optimal strategies to mitigate disruptions. The deterministic approach is particularly suitable for identifying critical vulnerabilities and optimal disruptions, as it finds more critical (combinations of) disruptions with less computational effort, especially for larger solution spaces. We conclude that **Obj 1** - Develop a valid and computationally tractable OAD model adapted for studying resilience in supply chains - has been achieved. The different experiments have also shown that subgoals **Obj 2** - Be able to explore a multitude of supply chain topologies and value chain complexities - and **Obj 3** - Be able to explore a multitude of disruption types and operational responses - are also achieved.

Our implementation has some limitations, as does any model. The resolution strategy of dualizing and collapsing the lower two levels in the sub-problem means that the operator problem should be a continuous LP. Nonlinear operator problems, which can model more realistic operational decisions, require a different resolution strategy, for example, by also using decomposition for the lower levels. Further research could investigate the use and resolution of non-LP continuous operator problems, which can model more interesting scenarios, albeit at a significant computational cost. We propose an important model extension for further research: extending the operator model to integrate a temporal dimension into the analysis of supply chain resilience, expanding beyond the current focus on spatial dynamics. By incorporating time, we can better understand how delays, expiration dates, and other temporal factors impact supply chains. The multi-commodity network flow will now be on a time-space decision graph, allowing for not only strategic-level decisions but also more tactical-level decisions. Such an operator model is highly likely to not be a continuous LP, further highlighting the importance of research into non-continuous LP operator problems. Finally, as discussed in the test case study, the model requires the user to define appropriate attack costs and budgets for the attacker sub-problem. We propose a method based on climate hazard data as a simple demonstration. However, further research could focus on developing more robust methods for determining attack budgets and costs, grounded in theoretical risk analysis. These costs will ultimately have a significant impact on how the model identifies critical parts of the supply chain and informs strategic design decisions.

References

Alderson, D. L., Brown, G. G., Carlyle, W. M., & Wood, R. K. (2011). Solving defender-attacker-defender models for infrastructure defense. *Center for Infrastructure Defense, Operations Research Department, Naval Postgraduate School*. https://doi.org/10.1287/ics.2011.0047

Aliakbar Hasani, S. H. Z., & Nikbakhsh, E. (2015). Robust closed-loop global supply chain network design under uncertainty: The case of the medical device industry. *International Journal of Production Research*, *53*(5), 1596–1624. https://doi.org/10.1080/00207543.2014.965349

Babazadeh, R., Razmi, J., Pishvaee, M. S., & Rabbani, M. (2017). A sustainable second-generation biodiesel supply chain network design problem under risk [New Research Frontiers in Sustainability]. *Omega*, *66*, 258–277. https://doi.org/https://doi.org/10.1016/j.omega.2015.12.010

Birge, J. R., Capponi, A., & Chen, P.-C. (2022). Disruption and rerouting in supply chain networks. *Operations Research*, *71*(2), 750–767. https://doi.org/10.1287/opre.2022.2409

Chakraborty, T., Chauhan, S. S., & Ouhimmou, M. (2019). Mitigating supply disruption with a backup supplier under uncertain demand: Competition vs. cooperation. *International Journal of Production Research*, *57*(12), 3618–3649. https://doi.org/10.1080/00207543.2019.1633025

Christopher, M., & Peck, H. (2004). Building the resilient supply chain [Copyright © 2004, Company]. *The International Journal of Logistics Management*, *15*(2), 1–14. https://doi.org/10.1108/09574090410700275

Fattahi, M., Govindan, K., & Keyvanshokooh, E. (2018). A multi-stage stochastic program for supply chain network redesign problem with price-dependent uncertain demands. *Computers & Operations Research*, *100*, 314–332. https://doi.org/https://doi.org/10.1016/j.cor.2017.12.016

Fattahi, M., Govindan, K., & Maihami, R. (2020). Stochastic optimization of disruption-driven supply chain network design with a new resilience metric. *International Journal of Production Economics*, *230*, 107755. https://doi.org/https://doi.org/10.1016/j.ijpe.2020.107755

Folke, C. (2006). Resilience: The emergence of a perspective for social–ecological systems analyses [Resilience, Vulnerability, and Adaptation: A Cross-Cutting Theme of the International Human Dimensions Programme on Global Environmental Change]. *Global Environmental Change*, *16*(3), 253–267. https://doi.org/https://doi.org/10.1016/j.gloenvcha.2006.04.002

Govindan, K., Fattahi, M., & Keyvanshokooh, E. (2017). Supply chain network design under uncertainty: A comprehensive review and future research directions [Production, Manufacturing and Logistics]. *European Journal of Operational Research*. www.elsevier.com/locate/ejor

Gupta, V., Ivanov, D., & Choi, T.-M. (2021). Competitive pricing of substitute products under supply disruption. *Omega*, *101*, 102279. https://doi.org/https://doi.org/10.1016/j.omega.2020.102279

Hamed Soleimani, M. S.-E., & Kannan, G. (2014). Incorporating risk measures in closed-loop supply chain network design. *International Journal of Production Research*, *52*(6), 1843–1867. https://doi.org/10.1080/00207543.2013.849823

Harrison, A., & Godsell, J. (2003). Responsive supply chains: An exploratory study of performance management.

Ho, W., Zheng, T., Yildiz, H., & Talluri, S. (2015). Supply chain risk management: A literature review [Authors' affiliations: University of Melbourne, Wayne State University, Michigan State University]. *International Journal of Production Research*, *53*(16). https://doi.org/10.1080/00207543.2015.1030467

Hosseini, S., Morshedlou, N., Ivanov, D., Sarder, M., Barker, K., & Khaled, A. A. (2019). Resilient supplier selection and optimal order allocation under disruption risks. *International Journal of Production Economics*, *213*, 124–137. https://doi.org/https://doi.org/10.1016/j.ijpe.2019.03.018

Huang, E., & Goetschalckx, M. (2014). Strategic robust supply chain design based on the pareto-optimal tradeoff between efficiency and risk. *European Journal of Operational Research*, *237*(2), 508–518. https://doi.org/https://doi.org/10.1016/j.ejor.2014.02.038

Huang, S., Lin, Y., Zhang, J., & Wang, P. (2023). Chance-constrained approach for decentralized supply chain network under uncertain cost [Research article]. *AIMS Mathematics*, *8*(5), 12217–12238. https://doi.org/10.3934/math.2023616

Karakatsoulis, G., & Skouri, K. (2023). A periodic review inventory model facing different disruption profiles. *International Journal of Production Economics*, *265*, 109004. https://doi.org/https://doi.org/10.1016/j.ijpe.2023.109004

Karakatsoulis, G., Skouri, K., & Lagodimos, A. (2024). Eoq with supply disruptions under different advance information regimes. *Applied Mathematical Modelling*, *125*, 772–788. https://doi.org/https://doi.org/10.1016/j.apm.2023.08.012

Kazemi Zanjani, M., Sanei Bajgiran, O., & Nourelfath, M. (2016). A hybrid scenario cluster decomposition algorithm for supply chain tactical planning under uncertainty. *European Journal of Operational Research*, *252*(2), 466–476. https://doi.org/https://doi.org/10.1016/j.ejor.2016.01.048

Keyvanshokooh, E., Ryan, S. M., & Kabir, E. (2016). Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated benders decomposition. *European Journal of Operational Research*, *249*(1), 76–92. https://doi.org/https://doi.org/10.1016/j.ejor.2015.08.028

Klibi, W., Martel, A., & Guitouni, A. (2010). The design of robust value-creating supply chain networks: A critical review. *European Journal of Operational Research*, *203*(2), 283–293. https://doi.org/https://doi.org/10.1016/j.ejor.2009.06.011

Kumar, S. K., & Tiwari, M. (2013). Supply chain system design integrated with risk pooling. *Computers & Industrial Engineering*, *64*(2), 580–588. https://doi.org/https://doi.org/10.1016/j.cie.2012.11.013

Li, Q., & Hu, G. (2014). Supply chain design under uncertainty for advanced biofuel production based on bio-oil gasification. *Energy*, *74*, 576–584. https://doi.org/https://doi.org/10.1016/j.energy.2014.07.023

Li, X., Kizito, R., & Paula, T. I. (2018). An agent-based simulation framework for supply chain disruptions and facility fortification, 821–832. https://doi.org/10.1109/WSC.2018.8632475

Liu, Y., & Guo, B. (2014). A lexicographic approach to postdisaster relief logistics planning considering fill rates and costs under uncertainty [Open Access, Academic Editor: Lu Zhen]. *Volume 2014*. https://doi.org/10.1155/2014/939853

Manners-Bell, J. (2014). *Supply chain risk: Understanding emerging threats to global supply chains*. Kogan Page Publishers.

Mousazadeh, M., Torabi, S., & Zahiri, B. (2015). A robust possibilistic programming approach for pharmaceutical supply chain network design. *Computers & Chemical Engineering*, *82*, 115–128. https://doi.org/https://doi.org/10.1016/j.compchemeng.2015.06.008

Mozafari, M., & Zabihi, A. (2020). Robust water supply chain network design under uncertainty in capacity. *Water Resources Management*, *34*, 4093–4112. https://doi.org/10.1007/s11269-020-02658-6

Pasandideh, S. H. R., Niaki, S. T. A., & Asadi, K. (2015). Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: Nsga-ii and nrga. *Information Sciences*, *292*, 57–74. https://doi.org/https://doi.org/10.1016/j.ins.2014.08.068

Pimentel, B. S., Mateus, G. R., & Almeida, F. A. (2013). Stochastic capacity planning and dynamic network design. *International Journal of Production Economics*, *145*(1), 139–149. https://doi.org/https://doi.org/10.1016/j.ijpe.2013.01.019

Qi, L. (2013). A continuous-review inventory model with random disruptions at the primary supplier. *European Journal of Operational Research*, *225*(1), 59–74. https://doi.org/https://doi.org/10.1016/j.ejor.2012.09.035

Rosenhead, J., Elton, M., & Gupta, S. K. (1972). Robustness and optimality as criteria for strategic decisions. *Journal of the Operational Research Society*, *23*(4), 20.

Ross, A. M., Rong, Y., & Snyder, L. V. (2008). Supply disruptions with time-dependent parameters [Part Special Issue: Topics in Real-time Supply Chain Management]. *Computers Operations Research*, *35*(11), 3504–3529. https://doi.org/https://doi.org/10.1016/j.cor.2007.01.018

Sheffi, Y. (2007). *The resilient enterprise: Overcoming vulnerability for competitive advantage* (Paperback). The MIT Press.

Shen, Z.-J. M., Coullard, C., & Daskin, M. S. (2003). A joint location-inventory model. *Transportation Science*, *37*, 40–55. https://doi.org/10.1287/trsc.37.1.40.12823

Snyder, L. V., Atan, Z., Peng, P., Rong, Y., Schmitt, A. J., & Sinsoysal, B. (2014). Or/ms models for supply chain disruptions: A review.

Soleimani, H., Govindan, K., Saghafi, H., & Jafari, H. (2017). Fuzzy multi-objective sustainable and green closed-loop supply chain network design. *Computers Industrial Engineering*, *109*, 191–203. https://doi.org/https://doi.org/10.1016/j.cie.2017.04.038

Subulan, K., Baykasoğlu, A., Özsoydan, F. B., Taşan, A. S., & Selim, H. (2015). A case-oriented approach to a lead/acid battery closed-loop supply chain network design under risk and uncertainty. *Journal of Manufacturing Systems*, *37*, 340–361. https://doi.org/https://doi.org/10.1016/j.jmsy.2014.07.013

Tanonkou, G.-A., Benyoucef, L., & Xie, X. (2008). Design of stochastic distribution networks using lagrangian relaxation. *IEEE Transactions on Automation Science and Engineering*, *5*(4), 597–608. https://doi.org/10.1109/TASE.2008.917156

Tomlin, B., & Wang, Y. (2011). *Operational strategies for managing supply chain disruption risk* (P. Kouvelis, L. Dong, O. Boyabatli, & R. Li, Eds.). Wiley. https://doi.org/10.1002/9781118115800.ch4

Tong, K., Gleeson, M. J., Rong, G., & You, F. (2014). Optimal design of advanced drop-in hydrocarbon biofuel supply chain integrating with existing petroleum refineries under uncertainty. *Biomass and Bioenergy*, *60*, 108–120. https://doi.org/https://doi.org/10.1016/j.biombioe.2013.10.023

Wang, D., Peng, J., Yang, H., Cheng, T., & Yang, Y. (2023). Distributionally robust location-allocation with demand and facility disruption uncertainties in emergency logistics. *Computers Industrial Engineering*, *184*, 109617. https://doi.org/https://doi.org/10.1016/j.cie.2023.109617

Wieland, A., & Durach, C. F. (2021). Two perspectives on supply chain resilience [[Article updated on March 17, 2021 after first online publication: reference to Rao et al. (2021) was corrected to Iyengar et al. (2021)]]. *Journal of Business Logistics*, *42*(3), 233–249. https://doi.org/10.1111/jbl.12271

Wieland, A., & Wallenburg, C. M. (2011). *Supply-chain-management in stürmischen zeiten*. Universitätsverlag der TU. Retrieved January 30, 2024, from https://www.gbv.de

Xiang, Y., & Wang, L. (2019). An improved defender–attacker–defender model for transmission line defense considering offensive resource uncertainties. *IEEE Transactions on Smart Grid*, *10*(3), 2534–2546. https://doi.org/10.1109/TSG.2018.2803783

Xu, J., Zhuang, J., & Liu, Z. (2016). Modeling and mitigating the effects of supply chain disruption in a defender–attacker game. *Annals of Operations Research*, *236*, 255–270. https://doi.org/10.1007/s10479-015-1810-z

Yılmaz Balaman, Ş., & Selim, H. (2014). A fuzzy multiobjective linear programming model for design and management of anaerobic digestion based bioenergy supply chains. *Energy*, *74*, 928–940. https://doi.org/https://doi.org/10.1016/j.energy.2014.07.073

Yongheng, J., Rodriguez, M. A., Harjunkoski, I., & Grossmann, I. E. (2014). Optimal supply chain design and management over a multi-period horizon under demand uncertainty. part ii: A lagrangean decomposition algorithm. *Computers & Chemical Engineering*, *62*, 211–224. https://doi.org/https://doi.org/10.1016/j.compchemeng.2013.11.014

Yuan, W., Zhao, L., & Zeng, B. (2014). Optimal power grid protection through a defender–attacker–defender model. *Reliability Engineering & System Safety*, *121*, 83–89. https://doi.org/10.1016/j.ress.2013.08.003

Zeballos, L. J., Méndez, C. A., Barbosa-Povoa, A. P., & Novais, A. Q. (2014). Multi-period design and planning of closed-loop supply chains with uncertain supply and demand [Selected papers from ESCAPE-23 (European Symposium on Computer Aided Process Engineering - 23), 9-12 June 2013, Lappeenranta, Finland]. *Computers & Chemical Engineering*, *66*, 151–164. https://doi.org/https://doi.org/10.1016/j.compchemeng.2014.02.027

Zhen, L., Zhuge, D., & Lei, J. (2016). Supply chain optimization in context of production flow network. *Journal of Systems Science and Systems Engineering*, *25*, 351–369. https://doi.org/10.1007/s11518-016-5304-6

Zokaee, S., Jabbarzadeh, A., Fahimnia, B., et al. (2017). Robust supply chain network design: An optimization model with real-world application [Published on November 30, 2014; Issue Date: October 2017]. *Ann Oper Res*, *257*, 15–44. https://doi.org/10.1007/s10479-014-1756-6

A

Scientific paper

The Operator-Disruptor-Resilience model for the design and operation of resilient supply chain networks.

Maurice Hart Nibbrig[a], Sh. Sharif Azadeh[b], M.Y. Maknoon[a,*]

[a] *Faculty of Technology, Policy, and Management, Delft University of Technology, 2628 BX, Delft, Netherlands*
[b] *Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN, Delft, Netherlands*

Abstract

This paper introduces the Operator-Disruptor-Resilience designer (**mODR**) model, a tri-level optimization model designed to enhance supply chain resilience against disruptions such as supplier failures and production shortfalls. The underlying supply chain conceptual model allows for both physical network as well as value chain flexibility. The model can model contingent rerouting and strategic resilience measures: increased capacity and multi-sourcing. The tri-level model is solved by being decomposed into an disruptor sub-problem and a master problem, solved iteratively. **mODR** efficiently solves realistically sized cases, as demonstrated in various computational experiments. The model's application to a pharmaceutical supply chain highlights its ability to integrate climate hazards into resilience planning. Comparative analysis with deterministic optimization confirms **mODR**'s effectiveness in identifying optimal disruptions, and proves to be a valuable tool for supply chain managers.

Keywords: Resilience, Supply Chain, Game Theory

1. Introduction

The global chip shortage of the early 2020s, intensified by geopolitical tensions and the COVID-19 pandemic, exposed significant vulnerabilities in global supply chains (SCs) and led to widespread production delays across various industries (Sweney, 2021). These events underscore the critical need for resilience in supply chain design and operation, as interconnected global SC networks face increasing exposure to a range of disruptions, from natural disasters to deliberate attacks. Ensuring the stability and integrity of these systems requires a deep understanding of disruption impacts and the integration of resilience strategies.

This research presents a novel approach to resilient supply chain design through the development of a three-level Mixed-Integer Programming (MIP) model, termed the Operator - Disruptor - Resilience designer (**mODR**) model. The **mODR** model integrates the roles of the operator, disruptor, and resilience designer into a comprehensive optimization framework, simulating potential disruptions and embedding both strategic and operational mitigation measures. This model provides actionable insights for enhancing supply chain resilience by addressing both natural and intentional threats. The disruptor component of the model allows for incorporating location-based climate risk data, which is demonstrated in a case study in Section 6.3 to introduce a framework for studying climate resilience.

Additionally, this research introduces a conceptual supply chain model that integrates both the physical network and the underlying value chain. Traditional supply chain design models, including those focused on resilience, typically address the physical network, such as location-allocation decisions, while assuming a fixed configuration of facility layers. However, the actual value chain—specifically the production chain, which encompasses the sequence of activities required to convert raw materials into finished products—can be complex, with multiple possible configurations. This research addresses this complexity by incorporating both aspects into the supply chain design process: the conceptual model allows for flexibility in both the physical network and the value chain.

The paper is structured as follows: Section 2 reviews the literature on supply chain resilience optimization. Section 3 describes the problem and introduces the supply chain conceptual model. Section 4 presents the **mODR** mathematical model, and Section 5 outlines the resolution strategy. Finally, Section 6 applies the model, including computational experiments, an analysis of system resilience through value chain and physical network flexibility, and a climate resilience case study.

2. Literature review

This section reviews the literature on resilience in supply chain network design and operations (SCNDO) problem. We discuss supply chain uncertainty and disruptions, and state-of-the-art resilience optimization models. We also briefly explore attacker-defender models, deterministic multi-level optimisation models found in various domains where resilience occurs.

A supply chain (SC) is a complex network of facilities that transform raw materials into finished products and distribute them to consumers (Harrison and Godsell, 2003). Supply chain

*Corresponding author
Email addresses: `maurice.hartn@gmail.com` (Maurice Hart Nibbrig), `s.sharifazadeh@tudelft.nl` (Sh. Sharif Azadeh), `M.Y.Maknoon@tudelft.nl` (M.Y. Maknoon)

management (SCM) focuses on optimizing this flow of goods for maximum efficiency (Wieland and Wallenburg, 2011). Supply chain network design (SCND) is a key SCM problem, involving the strategic configuration of production facilities, distribution centers, and transportation routes to enhance efficiency and meet objectives (Zhen et al., 2016). Supply chain resilience is defined as the capacity to persist, adapt, or transform in the face of change (Wieland and Durach, 2021). The concept has evolved from a focus on restoring the system to its original state (engineering resilience) to a more dynamic approach that emphasizes continuous adaptation (socio-ecological resilience) (Folke, 2006). SCND involves decisions under uncertainty, categorized into three types: (1) environments with known probability distributions, modeled using scenario approaches; (2) environments with unknown distributions, addressed with robust optimization; and (3) fuzzy environments with ambiguity and vagueness (Rosenhead et al., 1972). Studies on SCND models vary in the number of location layers, product types, material flows, and decision periods they consider, reflecting the complexity of real-world applications (Govindan et al., 2017).

Supply chain risks are categorized into operational risks, stemming from inherent uncertainties, and disruption risks, resulting from events like natural disasters (Ho et al., 2015). Resilient SCs are those that can operate efficiently both under normal and disruptive conditions. Strategies for managing disruptions include facility fortification, strategic stock management, and flexible sourcing (Snyder et al., 2014). Resilience is seen as essential in various contexts, including green and sustainable supply chains, as well as disaster relief operations (Govindan et al., 2017; Liu and Guo, 2014).

Optimizing SCND under uncertainty typically involves scenario-based stochastic programming, robust optimization, or fuzzy programming. These approaches balance between achieving optimal solutions and managing the complexity of large-scale optimization problems (Klibi et al., 2010; Hamed Soleimani and Kannan, 2014). However, challenges remain, such as generating representative scenarios and balancing conservatism in robust solutions (Fattahi et al., 2018; Keyvanshokooh et al., 2016). Disruptions, treated as a form of yield uncertainty, can be managed through strategies like optimizing inventory replenishment and flexible sourcing (Snyder et al., 2014). Facility location models also play a crucial role in disruption management, involving decisions on rerouting, fortification, and external collaboration (Wang et al., 2023; Li et al., 2018).

The attacker-defender model, a game theory-based approach, optimizes network resilience by integrating offensive (disruptive events) and defensive strategies into a unified framework. This model has been successfully applied in various domains, such as infrastructure and power systems, but remains underexplored in supply chain contexts (Alderson et al., 2011; Yuan et al., 2014; Xu et al., 2016).

The literature reveals gaps in addressing the interplay between mitigation and post-disruption strategies, as well as in integrating responses to both natural and deliberate disruptions. The attacker-defender model offers a promising approach to enhancing resilient supply chain network design (SCND) by providing deterministic insights and flexible adaptation, thereby addressing these gaps and advancing the field of supply chain resilience (Alderson et al., 2011; Govindan et al., 2017). Moreover, the reviewed models typically fix the physical layout configuration of the supply chain, such as the number of location layers. These models primarily focus on decisions related to the physical network, whereas, in reality, the network of production steps or the value chain can assume various forms. Models don't usually consider flexibility within the value chain in addition to physical network decisions.

3. Problem description

We consider a supply chain as a network of facilities that enable the flow of commodities through production, transformation, and distribution stages. Commodities, including raw materials, intermediates, and finished products, are transformed according to the *Bill of Materials* (BoM), which specifies the input quantities needed for each output unit. The network comprises four types of locations: *Suppliers*, *Producers*, *Warehouses*, and *Customers*. Locations are connected through a multi-modal transportation system, with each transport mode having its associated costs and capacity.

Supply chain resilience is defined as "the capacity of a supply chain to persist, adapt, or transform in the face of change" (Wieland and Durach, 2021). Disruptions introduce yield uncertainty, affecting the quantity of products delivered. These disruptions can disable production facilities (e.g., due to war), disrupt supplier processes (e.g., supply failures), affect producers (e.g., machine breakdowns), impair warehouse operations, or halt transportation networks (e.g., the Suez Canal closure). Strategies to counter disruptions include mitigation measures like flexible sourcing, multi-sourcing, and redundant production and storage capacity, as well as contingency actions such as post-disruption rerouting. The success of contingency actions heavily depends on prior mitigation decisions.

We aim to develop a framework for enhancing supply chain resilience in the face of disruptions. Our focus is on strategic decisions regarding supply chain design, evaluating resilience through an interdiction model. We model supply chain disruption as a Stackelberg game consisting of two opposing agents: the operator and the interdictor. The operator seeks to maximize performance, while the interdictor aims to minimize it. To improve supply chain resilinece, we introduce a third agent to allocate mitigation measures against disruptions. These agents interact in a tri-level optimization problem, resulting in strategies to enhance supply chain resilience.

In the following part, we outline our approach to developing a conceptual representation for the supply chain. This model will provide a baseline for analyzing and enhancing resilience against disruptions.

3.1. Conceptual model of supply chain operation

We model supply chain operations using two foundational concepts: the *physical network* and the *value chain*. The *physical* *network* depicts the handling and movement of commodities through various stages of the supply chain, including suppliers, manufacturers, warehouses, distribution centers, and end-users. In contrast, the *value chain* represents the sequence of activities required to convert raw materials into finished products delivered to customers.

Let $p \in P$ represent the set of *commodities*, encompassing items from raw materials to final products, and let $b \in B$ denote the set of conversion steps defined by *bills of materials* (BoMs). For each commodity p, we define three nodes: n_p^{Source} as the entry point of the commodity into the system, n_p^{Store} as the storage point for the commodity, and n_p^{Sink} as the point where the commodity reaches the end customer. Additionally, for each conversion process $b \in B$, we define the node $n_b^{Convert}$ to represent the conversion process based on the BoM. Each conversion process utilizes a set of inbound commodities to produce a set of outbound commodities.

We present the operational flow within the value chain using the *Production Graph* $G^{PROD} = \{N, E\}$, where $N = \{n_b^{Convert} \mid b \in B\} \cup \{n_p^{Source}, n_p^{Store}, n_p^{Sink} \mid p \in P\}$. The set of arcs E represent commodity flows between different production steps.

Example: Consider a value chain with six commodities,$P = \{RM1, RM2, RM3, IP1, IP2, FG\}$, and three production steps, $B = \{BoM1, BoM2, BoM3\}$. From these, we form the set of activity nodes N shown in Figure 1. The corresponding Production Graph $G^{PROD} = \{N, E\}$ is depicted in Figure 2.

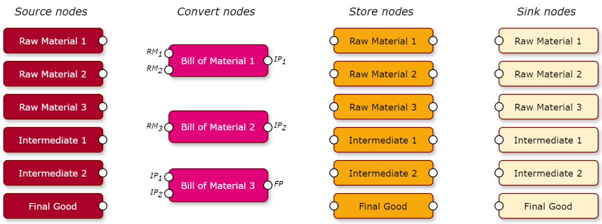


Figure 1: Example Production Graph nodes N. For each production step b, a `Convert` node $n_b^{Convert}$ is added. For each commodity p, `Source` n_p^{Source}, `Store` n_p^{Store}, and `Sink` n_p^{Sink} nodes are added. Circles represent possible connections for each node.

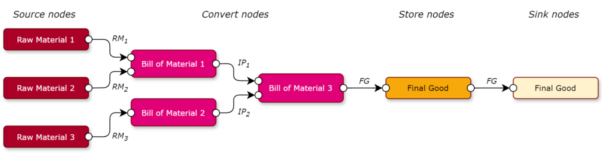


Figure 2: Example Production Graph $G^{PROD} = \{N, E\}$, formed by adding valid commodity flow arcs between relevant nodes.

Let L^S, L^P, L^W, and L^C be the sets of locations associated with *suppliers*, *producers*, *warehouses*, and *customers*, respectively. We define the graph $G^{PHYS} = \{L, \mathcal{E}\}$ to represent the physical network. Here, $L = L^S \cup L^P \cup L^W \cup L^C$ is the set of all locations, and $\mathcal{E}$ denotes the set of transport options (i.e., multiple arcs) between each pair of nodes, see Figure 3.

By integrating G^{PHYS} and G^{PROD}, we define the *Supply Chain Decision Network* as $G^{SC} = \{L, A\}$. Similar to G^{PHYS}, the set of nodes is based on the physical locations. The supply chain decision network maps suppliers, producers, warehouses, and customers to the corresponding `Source`, `Convert`, `Store`, and `Sink` nodes in the Production Graph. The set of arcs is created based on the movement of commodities between locations. It is important to note that this conceptual representation assesses supply chain resilience not only based on the physical movement of commodities but also includes various value chains involved in processing the products. Figure 4 shows the Supply Chain Decision Network of the illustrative example.

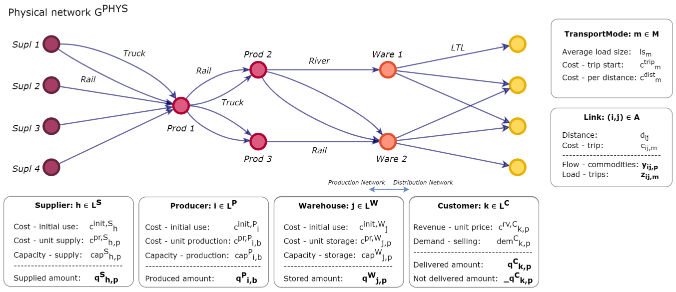


Figure 3: Illustration of the physical network of a supply chain $G^{PHYS} = \{L, \mathcal{E}\}$. It depicts supplier, producer, warehouse and customer locations and the transport links connecting them. Note that two locations may be connected by multiple different transport modes through a connection (i, j, m).

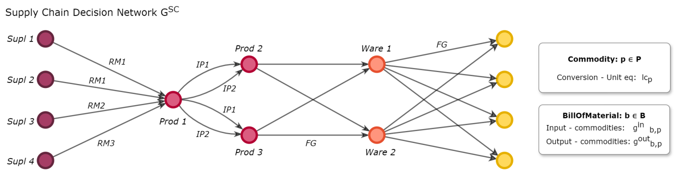


Figure 4: The resulting Supply Chain Decision Graph $G^{SC} = \{L, A^{flows}\}$. Multiple steps can be performed at a same facility, and multiple facilities can handle the same step. In our example, production steps `BoM1` and `BoM2` are performed at a single potential producer location `Prod1`, while `BoM3` is performed at two potential sites, `Prod2` and `Prod3`. Each arc in G^{SC} corresponds to a valid potential commodity flow between the locations.

4. Mathematical Formulation

We focus on the system design aspect of the supply chain and model resilience through the interactions between three key agents: the Operator (O), the Disruptor (D), and the Resilience Design (R). The *Operator* is responsible for managing the supply chain to ensure the efficient fulfillment of customer demand with the minimum cost. The *Disruptor* seeks to identify and target critical components of the supply chain to maximize disruptions, aiming to increase unsatisfied demand. The severity of these disruptions is controlled by the disruptor's budget, bdg^D. The *Resilience Design* is designed to anticipate potential disruptions and implement strategies to mitigate their impacts, thereby maintaining the stability and continuity of supply chain operations. The extent to which the system can be modified is governed by the resilience design budget, bdg^R.

In the optimization problem (4.1), the operator, disruptor, and resilience designer compete in a tri-level objective function to optimize the performance cost function Γ^O. The operator's decisions $\boldsymbol{x_O}$ are constrained by the valid system operations set X_O, which is influenced by the decisions of both the disruptor $\boldsymbol{x_D}$ and the resilience designer $\boldsymbol{x_R}$. The details of each agent's model are discussed in the following sections.

$$
\begin{aligned}
\min_{\boldsymbol{x_R}} \max_{\boldsymbol{x_D}} \min_{\boldsymbol{x_O}} \quad & \Gamma^O(\boldsymbol{x_O}, \boldsymbol{x_D}, \boldsymbol{x_R}) && (4.1)\\
\text{s.t.} \quad & \boldsymbol{x_O} \in X_O(\boldsymbol{x_D}, \boldsymbol{x_R}),\\
& \Gamma^D(\boldsymbol{x_D}) \leq bdg^D,\\
& \Gamma^R(\boldsymbol{x_R}) \leq bdg^R.
\end{aligned}
$$

4.1. Operator Problem

The operator's problem is formulated as a multi-commodity network flow problem on the graph G^{SC}. The vector of operational variables is $x_O = \begin{bmatrix} \boldsymbol{q^S} & \boldsymbol{q^P} & \boldsymbol{q^W} & \boldsymbol{q^C} & \boldsymbol{\bar{q}^C} & \boldsymbol{y} & \boldsymbol{z} \end{bmatrix}$, where $\boldsymbol{q^S}$, $\boldsymbol{q^P}$, and $\boldsymbol{q^W}$ represent decision variables for raw materials, production, and warehouse storage, respectively. Variables $\boldsymbol{q^C}$ and $\boldsymbol{\bar{q}^C}$ denote products delivered and not delivered to customers. The movement of commodities is denoted by $\boldsymbol{y}$, and $\boldsymbol{z}$ represents the number of transport trips between locations. A summary of the notations is provided in the Appendix.

The operator's problem includes constraints related to facility capacity, commodity continuity, production rates, and transportation. The detailed formulations of these constraints are provided below.

Facility Capacity. The facility capacity constraints (4.2) – (4.7) ensure that the quantities of commodities do not exceed the processing capacities of the respective facilities. Specifically, these constraints limit the amount q of raw materials p supplied at supplier h, the amount b processed at producer i, and the amount p stored at warehouse j to their respective capacity limits $cap^{\bullet}_{\bullet,\bullet}$. The constraints are defined as follows:

$$
\begin{aligned}
& q^S_{h,p} \leq cap^S_{h,p} && \forall h \in L^S, \forall p \in P && (4.2)\\
& q^P_{i,b} \leq cap^P_{i,b} && \forall i \in L^P, \forall b \in B && (4.3)\\
& q^W_{j,p} \leq cap^W_{j,p} && \forall j \in L^W, \forall p \in P && (4.4)\\
& q^S_{h,p} \geq 0 && \forall h \in L^S, \forall p \in P && (4.5)\\
& q^P_{i,b} \geq 0 && \forall i \in L^P, \forall b \in B && (4.6)\\
& q^W_{j,p} \geq 0 && \forall j \in L^W, \forall p \in P && (4.7)
\end{aligned}
$$

Continuity of Commodities. Constraints (4.8) – (4.14) ensure the movement of processed commodities between facilities, represented by the variable $y_{i,j,p}$. Let $n^-(i,p)$ denote the set of locations from which location i receives commodity p, and $n^+(i,p)$ denote the set of locations to which location i sends commodity p.

The supply constraints (4.8) ensure that raw materials $q^S_{h,p}$ supplied by a supplier h move to producers. The warehouse flow balance constraints (4.9) maintain the balance of commodities flowing into and out of warehouses j. Constraints (4.10) ensure the delivery of final products $q^C_{k,p}$ to customers k.

$$
\begin{aligned}
& q^S_{h,p} = \sum_{i \in n^+_{h,p}} y_{h,i,p} && \forall h \in L^S, \forall p \in P && (4.8)\\
& q^W_{j,p} = \sum_{i \in n^-_{j,p}} y_{i,j,p} = \sum_{k \in n^+_{j,p}} y_{j,k,p} && \forall j \in L^W, \forall p \in P && (4.9)\\
& q^C_{k,p} = \sum_{j \in n^-_{k,p}} y_{j,k,p} && \forall k \in L^C, \forall p \in P && (4.10)\\
& y_{i,j,p} \geq 0 && \forall (i,j) \in A, \forall p \in P(i,j) && (4.11)
\end{aligned}
$$

Constraints (4.12) ensure that the sum of the delivered amount $q^C_{k,p}$ and the unfulfilled amount $\bar{q}^C_{k,p}$ of final products equals the customer demand $dem^C_{k,p}$.

$$
\begin{aligned}
& q^C_{k,p} + \bar{q}^C_{k,p} = dem^C_{k,p} && \forall k \in L^C, \forall p \in P && (4.12)\\
& q^C_{k,p} \geq 0 && \forall k \in L^C, \forall p \in P && (4.13)\\
& \bar{q}^C_{k,p} \geq 0 && \forall k \in L^C, \forall p \in P && (4.14)
\end{aligned}
$$

Production Rate. The Bill of Materials (BoM) specifies all raw materials, sub-assemblies, and components required to produce an end product. In production processes, $g^{in}_{b,p}$ denotes the quantity of commodity p needed as input per unit of the final product according to BoM b, while $g^{out}_{b,p}$ denotes the quantity of commodity p produced as output per unit of the final product from BoM b.

Constraints (4.15) and (4.16) ensure the balance between input and output flows of commodities for each production process $q^P_{i,b}$. This balance is dictated by the BoM input and output requirements, represented by the terms $\sum y_{i,j,p}$.

$$
\begin{aligned}
& \sum_{b \in B} q^P_{i,b} \cdot g^{in}_{b,p} \leq \sum_{h \in n^-(i,p)} y_{h,i,p} && \forall i \in L^P, \quad \forall p \in P && (4.15)\\
& \sum_{b \in B} q^P_{i,b} \cdot g^{out}_{b,p} \geq \sum_{j \in n^+(i,p)} y_{i,j,p} && \forall i \in L^P, \quad \forall p \in P && (4.16)
\end{aligned}
$$

Transportation. The expected number of trips between each pair of locations is used to estimate the transportation cost. Let lc_p denote the standardized transportation unit for each commodity p and ls_m the average load size per trip using mode m. Constraints (4.17) estimate the number of trips required with mode m between each pair of locations $(i,j) \in A$.

$$
\begin{aligned}
& \sum_{p \in P(i,j)} y_{i,j,p} \cdot lc_p \leq \sum_{m \in M(i,j)} z_{i,j,m} \cdot ls_m && \forall (i,j) \in A && (4.17)\\
& z_{i,j,m} \geq 0\\
& \qquad \forall (i,j) \in A, \forall m \in M(i,j) && && (4.18)
\end{aligned}
$$

4.2. Disruption Problem

Supply chain vulnerability is defined as the set of simultaneous disruptions in the system. The disruption variables vector is $\boldsymbol{x_A} = \begin{bmatrix} \boldsymbol{\phi^S} & \boldsymbol{\phi^P} & \boldsymbol{\phi^W} & \boldsymbol{\psi^S} & \boldsymbol{\psi^P} \end{bmatrix}$, where $\boldsymbol{\phi^S}$, $\boldsymbol{\phi^P}$, and $\boldsymbol{\phi^W}$ are binary variables indicating full disablement of a supplier, producer, or warehouse, respectively. Partial disruptions to specific processes at facilities are represented by $\boldsymbol{\psi^S}$ and $\boldsymbol{\psi^P}$. These partial disruptions are modeled using disruption impact levels $f \in F$, which represent a loss of capacity. The list of notations is provided in the appendix.

Let u_f represent the percentage of capacity lost due to a partial failure. Using these variables and parameters, the disruption problem impacts the facility's capacity. Consequently, constraints (4.2)–(4.4) are revised to model both partial and complete disruptions. The modified constraints are:

$$q^S_{h,p} \leq cap^S_{h,p} \cdot \min(1 - \phi^S_h, 1 - u_f \cdot \psi^S_{h,p,f}) \quad \forall f \in F, \forall h \in L^S, \forall p \in P \tag{4.2}$$

$$q^P_{i,b} \leq cap^P_{i,b} \cdot \min(1 - \phi^P_i, 1 - u_f \cdot \psi^P_{i,b,f}) \quad \forall f \in F, \forall i \in L^P, \forall b \in B \tag{4.3}$$

$$q^W_{j,p} \leq cap^W_{j,p} \cdot \min(1 - \phi^W_j) \qquad \forall j \in L^W, \forall p \in P \tag{4.4}$$

$$\phi^S_h \in \{0,1\} \quad \forall h \in L^S \tag{4.19}$$

$$\phi^P_i \in \{0,1\} \quad \forall i \in L^P \tag{4.20}$$

$$\phi^W_j \in \{0,1\} \quad \forall j \in L^W \tag{4.21}$$

$$\psi^S_{h,p,f} \in \{0,1\} \quad \forall h \in L^S, \forall p \in P, \forall f \in F \tag{4.22}$$

$$\psi^P_{i,b,f} \in \{0,1\} \quad \forall i \in L^P, \forall b \in B, \forall f \in F \tag{4.23}$$

The min function handles multiple capacity-constraining terms. The term $q^{\cdot}_{i,p} \leq cap^{\cdot}_{i,p}(1 - \phi^{\cdot}_i)$ reduces facility i's capacity for process p to zero if fully disabled by $\phi^{\cdot}_i$. For each disruption impact level f, the term $q^{\cdot}_{i,p} \leq cap^{\cdot}_{i,p}(1 - u_f \cdot \psi^{\cdot}_{i,p,f})$ models capacity reduction due to the partial process disruption variable $\psi^{\cdot}_{i,p,f}$.

Finally, we introduce Γ^D and bdg^D as the disruptor's disruption costs and budget, respectively. This budget controls the severity of disruptions. Constraint (4.24) ensures that disruption severity remains within the budget bdg^D.

$$\begin{aligned}\Gamma^D = &\sum_{h \in L^S} c^{D,loc}_h \cdot \phi^S_h + \sum_{i \in L^P} c^{D,loc}_i \cdot \phi^P_i + \sum_{j \in L^W} c^{D,loc}_j \cdot \phi^W_j \\ &+ \sum_{f \in F} \left(\sum_{h \in L^S} \sum_{p \in P} c^{D,pr,S}_{h,p,f} \cdot \psi^S_{h,p,f} + \sum_{i \in L^P} \sum_{b \in B} c^{D,pr,P}_{i,b,f} \cdot \psi^P_{i,b,f} \right) \\ \leq \quad & bdg^D \end{aligned} \tag{4.24}$$

Total disruption costs Γ^D include the costs of fully disabling facilities, represented by $c^{D,loc}_i \cdot \phi^{\cdot}_i$ for each facility i. They also include the costs of partially or fully disrupting specific supply and production processes at a facility, modeled by $c^{D,pr,\cdot}_{i,\cdot,f} \cdot \psi^{\cdot}_{i,\cdot,f}$ for each process p or b at each facility i and for each disruption impact level f. The total disruption costs must remain within the available budget.

4.3. Resilience Design Problem

The Resilience Design encompasses system design actions such as flexible sourcing, multi-sourcing, and redundant production and storage capacity. These actions are represented by the vector $\boldsymbol{x_D} = \begin{bmatrix} \boldsymbol{x^S} & \boldsymbol{x^P} & \boldsymbol{x^W} \end{bmatrix}$, where $\boldsymbol{x^S}$, $\boldsymbol{x^P}$, and $\boldsymbol{x^W}$ are binary decisions regarding the establishment of facility locations for suppliers h, producers i, and warehouses j. Each decision enhances the network's capacity to manage disruptions and maintain operations under diverse conditions, thereby increasing the overall resilience of the system.

To incorporate these design decisions, we modify the facility capacity constraints (4.2)–(4.4). The revised constraints ensure that a facility's capacity for process p can only be utilized if the facility is included in the system design through the binary variable $x^{\cdot}_i$. The modified constraints are:

$$q^S_{h,p} \leq cap^S_{h,p} \cdot \min(x^S_h, 1 - \phi^S_h, 1 - u_f \cdot \psi^S_{h,p,f}) \quad \forall f \in F, \forall h \in L^S, \forall p \in P \tag{4.2}$$

$$q^P_{i,b} \leq cap^P_{i,b} \cdot \min(x^P_i, 1 - \phi^P_i, 1 - u_f \cdot \psi^P_{i,b,f}) \quad \forall f \in F, \forall i \in L^P, \forall b \in B \tag{4.3}$$

$$q^W_{j,p} \leq cap^W_{j,p} \cdot \min(x^W_j, 1 - \phi^W_j) \qquad \forall j \in L^W, \forall p \in P \tag{4.4}$$

$$x^S_h \in \{0,1\} \qquad \forall h \in L^S \tag{4.25}$$

$$x^P_i \in \{0,1\} \qquad \forall i \in L^P \tag{4.26}$$

$$x^W_j \in \{0,1\} \qquad \forall j \in L^W \tag{4.27}$$

Additionally, the operator is subject to a budget constraint, ensuring that the total system design costs Γ^R do not exceed the budget bdg^R. The total cost is the sum of the initial investment costs $c^{init,\cdot}_i \cdot x^{\cdot}_i$ for establishing each supplier h, producer i, and warehouse j.

$$\Gamma^R = \sum_{h \in L^S} c^{init,S}_h \cdot x^S_h + \sum_{i \in L^P} c^{init,P}_i \cdot x^P_i + \sum_{j \in L^W} c^{init,W}_j \cdot x^W_j \leq bdg^R \tag{4.28}$$

4.4. Tri-level Operator-Disruption-Resilience Model

The **mODR** (Operator-Disruption-Resilience) model integrates the operator, disruptor, and resilience design problems into a tri-level optimization problem. The objective function (4.29), Γ^O, evaluates the system's operational performance, which the operator aims to minimize and the disruptor aims to maximize.

[mODR]

$$\min_{x_R} \max_{x_D} \min_{x_O} \quad \Gamma^O = \rho^c \cdot \Gamma^{costs} + \rho^r \cdot \Gamma^{penal} \tag{4.29}$$

s.t.

Operator Constraints:	(4.2) – (4.14)
Disruptor Constraints:	(4.24) – (4.23)
Resilience Designer Constraints:	(4.28) – (4.27)

The supply chain performance, Γ^O, is defined as a function of operational costs, Γ^{costs}, and penalty costs, Γ^{penal}, due to unmet customer demand. The parameters ρ^c and ρ^r, where $\rho^c + \rho^r = 1$, balance the importance of these costs. The penalty costs, Γ^{penal}, account for the penalties associated with non-delivery of final products p to customers k. Instead of imposing strict demand-satisfaction constraints, we incorporate non-delivery penalties in the objective function to better model supply chain resilience. These penalty costs are calculated based on the non-delivered quantities $\bar{q}^C_{k,p}$ and the associated penalty costs $c^{rv,C}_{k,p}$ (representing lost revenue), using:

$$\Gamma^{penal} = \sum_{k \in L^C} \sum_{p \in P} c^{rv,C}_{k,p} \cdot \bar{q}^C_{k,p} \tag{4.30}$$

Operational costs, Γ^{costs}, are decomposed into fixed operational costs Γ^{fixed}, process costs Γ^{proc}, and transport costs Γ^{trans}:

$$\Gamma^{costs} = \Gamma^{fixed} + \Gamma^{proc} + \Gamma^{trans} \tag{4.31}$$

Fixed Operational Costs. The fixed operational costs, Γ^{fixed}, include expenses such as rent and administrative costs for suppliers, producers, and warehouses. Let x^S_h, x^P_i, and x^W_j be binary variables indicating the utilization of supplier h, producer i, and warehouse j, respectively. These costs are calculated as:

$$\Gamma^{fixed} = \sum_{h \in L^S} c^{fix,S}_h \cdot x^S_h + \sum_{i \in L^P} c^{fix,P}_i \cdot x^P_i + \sum_{j \in L^W} c^{fix,W}_j \cdot x^W_j \tag{4.32}$$

Process Costs. Process costs, Γ^{proc}, are incurred during the supply, production, and storage stages. Let $q^S_{h,p}$ be the amount of commodity p supplied by supplier h, $q^P_{i,b}$ be the amount of bill of material b produced at producer i, and $q^W_{j,p}$ be the amount of commodity p stored at warehouse j. The process costs are:

$$\Gamma^{proc} = \sum_{h \in L^S} \sum_{p \in P} c^{pr,S}_{h,p} \cdot q^S_{h,p} + \sum_{i \in L^P} \sum_{b \in B} c^{pr,P}_{i,b} \cdot q^P_{i,b} + \sum_{j \in L^W} \sum_{p \in P} c^{pr,W}_{j,p} \cdot q^W_{j,p} \tag{4.33}$$

Transport Costs. Transport costs, Γ^{trans}, are associated with the movement of goods between locations. Let $z_{i,j,m}$ denote the number of trips between locations i and j using transport mode m. Each trip incurs a cost $c^{tr}_{i,j,m}$, which includes a fixed trip start cost c^{trip}_m and a variable distance-based cost $c^{dist}_m \cdot d_{i,j}$. The transport costs are:

$$\Gamma^{trans} = \sum_{(i,j) \in A} \sum_{m \in M(i,j)} (c^{trip}_m + c^{dist}_m \cdot d_{i,j}) \cdot z_{i,j,m} \tag{4.34}$$

5. Resolution Approach

In Section 4, the supply chain resilience problem is formulated as a tri-level optimization problem (**mODR**) with hierarchical decision-making. In this section, we present a decomposition approach inspired by the methods of Alderson et al. (2011) and Ghorbani-Renani et al. (2021). The core idea of our approach is to decompose the tri-level **mODR** model into two parametrized models: a single-level master problem (**mODR-Master**), which provides a lower bound, and a bi-level disruptor sub-problem (**mOD-Sub**), which determines the upper bound on the solution of the tri-level optimization.

Given a disruption vector $\hat{\boldsymbol{x}}_{\boldsymbol{Dk}}$, there exists a corresponding operational response $\hat{\boldsymbol{x}}_{\boldsymbol{Ok}}$ forming a pair $(\hat{\boldsymbol{x}}_{\boldsymbol{Dk}}, \hat{\boldsymbol{x}}_{\boldsymbol{Ok}})$. This allows the **mODR** to be decomposed into a relaxed master problem (**mODR-Master**) and a disruptor sub-problem (**mOD-Sub**).

For clarity, vectors with a hat (e.g., $\hat{\boldsymbol{x}}_{\boldsymbol{R}}$) represent fixed parameters. Vectors without a hat ($\boldsymbol{x}_{\boldsymbol{R}}$) are decision variables, and vectors with a star ($\boldsymbol{x}_{\boldsymbol{R}}{}^*$) denote optimal values.

The **mODR-Master** yields an optimal system design $\boldsymbol{x}_{\boldsymbol{R}}{}^*$ for a given set of potential disruptions and the associated optimal disruption-response pair $(\boldsymbol{x}_{\boldsymbol{D}}{}^*, \boldsymbol{x}_{\boldsymbol{O}}{}^*)$. Conversely, the **mOD-Sub** determines the optimal disruption vector $\boldsymbol{x}_{\boldsymbol{D}}{}^*$ for a given system design $\hat{\boldsymbol{x}}_{\boldsymbol{R}}$.

Our algorithm iteratively generates the disruption vectors $\hat{\boldsymbol{x}}^{\boldsymbol{K}}_{\boldsymbol{D}} = \{\hat{\boldsymbol{x}}_{\boldsymbol{D1}}, ..., \hat{\boldsymbol{x}}_{\boldsymbol{Dk}}\}$ instead of enumerating all possible disruption vectors. The solution approach involves solving the sub-problem to update the disruption vector subset. Then, the master problem is re-solved for the optimal design. This process iterates until convergence.

The decomposition approach is executed through an iterative algorithm, detailed in the pseudo-code in 1. The algorithm begins with the initialization of the full input data required by **mODR**, an optimality gap ε (where $0 \leq \varepsilon < 1$), and a maximum number of iterations K_{MAX}. During each iteration, the disruptor sub-problem updates the upper bound on the optimal operational costs, denoted as z^{UP}. Concurrently, the master problem updates the lower bound, denoted as z^{LO}. The algorithm concludes when the normalized difference between these bounds is less than ε.

Algorithm 1: mODR Iterative Resolution

Input: Optimality tolerance ε, maximum iterations K_{MAX}

Output: $(\boldsymbol{x}_{\boldsymbol{R}}{}^*, \boldsymbol{x}_{\boldsymbol{D}}{}^*, \boldsymbol{x}_{\boldsymbol{O}}{}^*)$

Initialize an empty set of disruption vectors $\hat{\boldsymbol{x}}_{\boldsymbol{D}}{}^{\boldsymbol{0}} \leftarrow \{\emptyset\}$;
Select an initial feasible disruption vector $\hat{\boldsymbol{x}}_{\boldsymbol{D0}}$ (e.g., "no disruption");
Solve **mOD**($\hat{\boldsymbol{x}}_{\boldsymbol{D0}}$) for optimal $\boldsymbol{x}_{\boldsymbol{R}1}$ and z^*;
Initialize $z^{LO} \leftarrow z^*, z^{UP} \leftarrow +\infty, K \leftarrow 1$;
while $z^{UP} - z^{LO} > |z^{LO}| \cdot \varepsilon$ ***and*** $K < K_{MAX}$ **do**
  Solve **mOD-Sub**($\hat{\boldsymbol{x}}_{\boldsymbol{RK}}$) for $\boldsymbol{x}_{\boldsymbol{DK}}$ and z;
  if $z < z^{UP}$ **then**
    $\boldsymbol{x}_{\boldsymbol{R}}{}^* \leftarrow \hat{\boldsymbol{x}}_{\boldsymbol{RK}}; \boldsymbol{x}_{\boldsymbol{D}}{}^* \leftarrow \boldsymbol{x}_{\boldsymbol{DK}}; z^{UP} \leftarrow z$;
  end
  Add $\boldsymbol{x}_{\boldsymbol{DK}}$ to the set of attacks: $\hat{\boldsymbol{x}}_{\boldsymbol{D}}{}^{\boldsymbol{K}} \leftarrow \hat{\boldsymbol{x}}_{\boldsymbol{D}}{}^{\boldsymbol{K}} \cup \{\boldsymbol{x}_{\boldsymbol{DK}}\}$;
  Solve **mODR-Master**($\hat{\boldsymbol{x}}_{\boldsymbol{D}}{}^{\boldsymbol{K}}$) for $\boldsymbol{x}_{\boldsymbol{R}K+1}$ and z;
  if $z > z^{LO}$ **then**
    $z^{LO} \leftarrow z$;
  end
  $K \leftarrow K + 1$;
end
Solve **mO**($\boldsymbol{x}_{\boldsymbol{R}}{}^*, \boldsymbol{x}_{\boldsymbol{D}}{}^*$) for $\boldsymbol{x}_{\boldsymbol{O}}{}^*$;
return $(\boldsymbol{x}_{\boldsymbol{R}}{}^*, \boldsymbol{x}_{\boldsymbol{D}}{}^*, \boldsymbol{x}_{\boldsymbol{O}}{}^*)$;

5.1. Master Problem *mODR-Master*

mODR-Master optimizes the system design $\boldsymbol{x_R}^*$ given a subset of possible disruption vectors $\hat{\boldsymbol{x}}^{\boldsymbol{K}}_{\boldsymbol{D}}$. Besides the design variables, **mODR-Master** includes Z, a continuous variable representing the system performance, and operational variables $\boldsymbol{x_O}_1, ..., \boldsymbol{x_O}_K$ for each disruption vector. The design variables must satisfy the designer constraints, noted as (5.2), where X_R denotes the set of valid design vectors $\boldsymbol{x_R}$.

[mODR-Master]

$$\min_{Z,\boldsymbol{x_R},\boldsymbol{x_O}_1,...,\boldsymbol{x_O}_K} \quad Z \tag{5.1}$$

$$\text{s.t.} \quad \boldsymbol{x_R} \in X_R \tag{5.2}$$

$$\boldsymbol{x_O}_k \in X_O(\boldsymbol{x_R}, \hat{\boldsymbol{x_D}}_k) \quad \forall k \in K \tag{5.3}$$

$$Z \geq \Gamma^{total}(\boldsymbol{x_R}, \hat{\boldsymbol{x_D}}_k, \boldsymbol{x_O}_k) \quad \forall k \in K \tag{5.4}$$

Operational variables $\boldsymbol{x_O}_1, ..., \boldsymbol{x_O}_K$ must satisfy all operational constraints (4.2) - (4.17) for each disruption vector $\hat{\boldsymbol{x_D}}_1, ..., \hat{\boldsymbol{x_D}}_K$, as expressed in (5.3), where $X_O(\boldsymbol{x_R}, \hat{\boldsymbol{x_D}}_k)$ denotes the set of valid operational variables for design $\boldsymbol{x_R}$ and setting $\hat{\boldsymbol{x_D}}_k$. Finally, constraints (5.4) ensure that the objective value of the master problem is at least the operational cost associated with each disruption-response pair $(\hat{\boldsymbol{x_D}}_k, \hat{\boldsymbol{x_O}}_k)$.

5.2. Disruptor Sub-Problem *mOD-Sub*

To solve the bi-level disruptor sub-problem **mOD-Sub**, we employ a "dualize-and-combine" approach based on the duality theory presented in Dempe and Zemkoho (2020). In this problem, the operator model is a linear programming problem. As a result, the bi-level sub-problem **mOD-Sub** is reformulated as a single-level quadratic problem (named **mOD-MIQP**) by replacing the inner minimization problem with its dual maximization problem. The solution identifies the optimal disruption vector $\boldsymbol{x_D}^*$, which maximizes the damage to the operator's performance function, given a predetermined system design $\hat{\boldsymbol{x_R}}$. The full sub-problem **mOD-MIQP** definition is presented in the Appendix Appendix B.

6. Results

All experiments were conducted using Python and Gurobi 11.0.2 on a Core i5 processor with 16GB of RAM. In Section 6.1, we evaluate the computational performance of our approach, demonstrating its efficiency in solving a wide range of realistically sized cases within a reasonable timeframe. In Section 6.2, we analyze network resilience by varying flexibility in the value chain and the physical network, finding that incorporating value chain flexibility significantly enhances system resilience within the same design budget. Finally, in Section 6.3, we present a resilience analysis of a real-life pharmaceutical supply chain, showing how our model integrates climate hazards into the resilience modelling.

Instance Definition. A supply chain instance consists of a value chain and physical network configuration. The value chain details the sequence and interconnectivity of production steps within the supply chain, represented as a Production Graph G^{PROD}. The physical network specifies the number of facilities, such as suppliers, producers, warehouses, and end-customers.

Scenarios. A scenario is defined by a disruptor budget bdg^D and a system design budget bdg^R, representing the resources available to the disruptor and designer, respectively. System design budgets bdg^R are represented in monetary units, with existing facilities initially costing zero. Disruptor budgets bdg^D are in the same unit as disruption costs $c^{D,\cdot}$. A facility i can be fully disrupted at cost $c_i^{D,loc}$, or a process p at a facility i can be partially disrupted at cost $c_{i,p,f}^{D,pr}$, where $f \in F$ is the disruption impact level. The disruption impact levels are defined as $F =$ `Minor` (-10% capacity), `Heavy` (-20%), `Major` (-50%), `Fatal` (-100%).

6.1. Computational Performance

To assess the computation performance of the proposed resolution approach, we generated twelve instances of supply chain networks: 12 combinations of three possible value chain layouts and four physical network sizes.

Value Chain Layout. We define the following four generic randomly generated value chains. The details of these are presented in Table 1 and illustrated in Figure 5.

- `VC-Simple`: A straightforward single-step process utilizing two raw materials to produce one final product, typical of basic food production.
- `VC-LinSingl`: A linear sequence involving multiple production steps that transform two raw materials into a single final product, common in elementary manufacturing settings.
- `VC-LinMulti`: Multiple parallel linear processes that generate several final products, representing diversified manufacturing operations.
- `VC-Complex`: A complex network characterized by numerous inputs and outputs, similar to those found in automotive manufacturing.

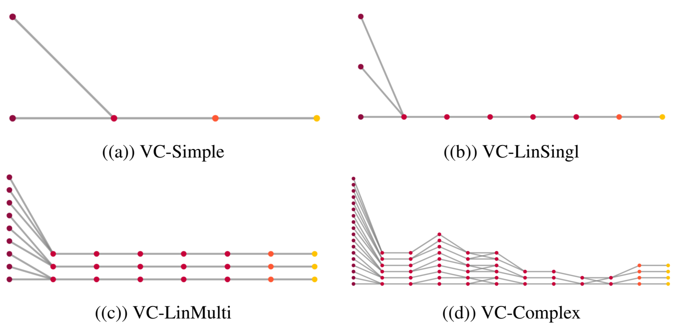


Figure 5: Production graphs of the different value chains.

Physical Network. To evaluate the impact of decision network size on computational performance, we vary the physical network's size, which in turn influences the potential sys-

Table 1: Computational performance instance definition - Value chains

Value chain	# Production steps	# Commodities
`VC-Simple`	1	3
`VC-LinSingl`	5	8
`VC-LinMulti`	15	24
`VC-Complex`	43	74

tem design configurations and the operational responses to disruptions. We categorize the network into three sizes: `Small`, `Medium`, and `Large`, representing the decision network size of a localized supply chain up to a global-scale supply chain with numerous facilities and demand points. For each network size, locations and capacities are randomly assigned to reflect the realistic variability and complexity inherent in supply chain management. Table 2 outlines the specific characteristics of each physical network configuration.

Table 2: Computational performance instance definition - Decision network sizes

Size	# Suppliers	# Producers	# Warehouses	# Customers
`Small`	10	5	5	10
`Medium`	30	15	15	50
`Large`	50	25	25	200

Scenarios. For each configuration, we assess the computational performance by varying the disruption and resilience budgets. In contrast to the following experiments where the budgets are presented in monetary terms, for the computation experment we present the budgets in relative terms, focusing on their relative influence rather than absolute values. Introducing a new supplier incurs a relatively low initial cost, whereas establishing a new producer entails a significantly higher expense. The cost for a new warehouse falls somewhere in between. The table below explains the value for each unit in terms of disruption impact levels and the corresponding number of new facilities that can be established within each resilience budget. Details of each budget are presented in Table 3.

Table 3: Scenarios - Disruptor and Design Budgets and their relative resource value.

	Disruption Level					New Facilities		
Γ^D	Minor	Heavy	Major	Fatal	Γ^R	# Suppliers	# Producers	# Warehouses
None	0	0	0	0	**None**	0	0	0
Mild	1	0	0	0	**Minimal**	1	0	0
Moderate	25	5	1	0	**Moderate**	10	0	1
Severe	50	12	2	0	**Extensive**	100	1	10
Catastrophic	200	50	8	2	**Global**	500	5	50

Computational performance analysis. As presented in the tables below, the computational experiments demonstrate that the model is solvable within a reasonable time frame for a wide range of realistically sized cases. For all runs, the minimum gap ε was set to 10^{-5}, and the maximum computation time was set to 90 minutes (5400 seconds). Overall, 291 cases out of 300 were solved within the selected time limit. As can be seen in these tables, three factors contribute to the computational performance of our model: (1) network size, (2) the budget of disruption, and (3) the budget considered for resilience design. By increasing the network size, we increase the number of design decisions ($\mathbf{x_R}$), which enlarges the size of the **mODR-Master** problem, considered the performance bottleneck of the algorithm, see mean solve time in Table 5. On the other hand, the complexity of the value chain and its production graph does not directly correlate with solve time. Although the complex value chain introduces additional operational settings (i.e., $\mathbf{x_D}$), this does not necessarily lead to longer solution times, see Table 4.

Additionally, increasing the budgets for disruption and resilience design (i.e., bdg^D and bdg^R) affects the solve time. In general, higher budgets result in increased computational time, see Tables 6 and 7. The main reason is that higher budget values result in fewer constraints on $\mathbf{x_D}$ and $\mathbf{x_R}$, which ultimately increases the possible pool of solutions. On the other hand, with lower budgets, the model discards solutions more easily. The full results are reported in Appendix Appendix C.

Tables 4 through 7 present the impact of individual variables on the mean, minimum, and maximum solve times and the final gap. Table 8 provides these results for each specific instance, i.e. each combination of value chain and network size. Comprehensive results for each scenario, and for each combination of disruption and design budget, are detailed in Tables C.18 to C.22 in Appendix Appendix C.

Table 4: Value chain

	Simple	Linear (single)	Linear (multi)	Complex
Time [s], min	0.00	0.01	0.01	0.01
Time [s], mean	260.30	518.60	869.80	161.29
Time [s], max	5400.00	5400.00	5400.00	5400.00
Gap [%], min	0.00	0.00	0.00	0.00
Gap [%], mean	4.65	4.79	6.71	0.00
Gap [%], max	174.23	103.23	221.10	0.00

Table 5: Computational performance - Network size

	Small	Medium	Large
Time [s], min	0.00	0.01	0.04
Time [s], mean	0.86	353.98	1002.66
Time [s], max	5.34	5400.00	5400.00
Gap [%], min	0.00	0.00	0.00
Gap [%], mean	0.00	3.12	8.99
Gap [%], max	0.00	173.64	221.10

Table 6: Computational performance - Disruption budget bdg^D

	None	Mild	Moderate	Severe	Catastrophic
Time [s], min	0.00	0.02	0.03	0.04	0.03
Time [s], mean	0.33	300.53	674.01	441.43	846.17
Time [s], max	2.73	5400.00	5400.00	5400.00	5400.00
Gap [%], min	0.00	0.00	0.00	0.00	0.00
Gap [%], mean	0.00	0.36	2.20	2.40	15.21
Gap [%], max	0.00	10.98	66.13	108.71	221.10

Table 7: Computational performance - Design budget bdg^R

	None	Minimal	Moderate	Extensive	Global
Time [s], min	0.00	0.01	0.01	0.01	0.01
Time [s], mean	0.54	2.87	121.18	860.94	1276.96
Time [s], max	3.34	16.78	4290.80	5400.00	5400.00
Gap [%], min	0.00	0.00	0.00	0.00	0.00
Gap [%], mean	0.00	0.00	0.00	9.36	10.81
Gap [%], max	0.00	0.00	0.00	174.23	221.10

Table 8: Computational performance - Instances

Value chain	VC-Simple			VC-LinSingl			VC-LinMulti			VC-Complex		
Network size	Small	Medium	Large	Small	Medium	Large	Small	Medium	Large	Small	Medium	Large
time mean	0.42	312.62	467.87	0.77	315.18	1239.88	0.55	785.08	1823.77	1.72	3.02	479.14
time min	0.00	0.01	0.04	0.01	0.01	0.06	0.01	0.02	0.09	0.01	0.08	0.17
time max	5.34	5400.00	5400.00	4.35	5400.00	5400.00	2.39	5400.00	5400.00	3.82	7.66	5400.00
gap mean	0.00	6.95	6.99	0.00	4.11	10.25	0.00	1.41	18.73	0.00	0.00	0.00
gap min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gap max	0.00	173.64	174.23	0.00	102.87	103.23	0.00	35.29	221.10	0.00	0.00	0.00

6.2. *Value chain-based network analysis*

In this subsection, we investigate a hypothetical steel manufacturing supply chain and the impact of considering both value chain and physical network flexibility on supply chain resilience.

Instance definition.We define four instances that differ in their value chain and physical network flexibility:

I_0 - Single path production chain, no physical flexibility (base case);
I_1 - Single path production chain, flexible physical network;
I_2 - Multiple paths production chain, no physical flexibility;
I_3 - Multiple paths production chain, flexible physical network.

Instance I_0. In this base case instance, we explore a steel manufacturing supply chain without flexibility in its production chain or physical network. The fictional steel manufacturing process takes `crude coal` and `crude iron` as initial input commodity, and through a linear chain of transformation processes, produces `finished goods`, delivered to customers. The production steps and the single commodity flow path are shown in Figure 6.

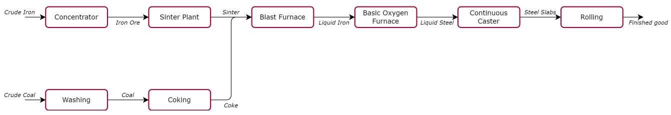


Figure 6: Production graph with a single possible commodity flow path, for instances I_0 and I_1. Only the `Convert` (bill of material) nodes are shown.

The physical network consists of the following open locations. Their positions are randomized.

2 suppliers - providing `crude coal` and `crude iron`, respectively;
4 producers - handling various stages of production;
10 warehouses - storing `finished goods`;
100 customers - demanding `finished goods`.

Instance I_1. Instance I_1 introduces physical network flexibility by including additional potential facilities. The expanded decision network includes 10 potential suppliers, 16 producers, and 30 warehouses. The number of customers remain constant.

Instance I_2. Instance I_2 adds flexibility to the production chain by introducing alternative production paths. We add a main production path which includes `iron pellets`, `direct reduced iron` (DRI), and `electric arc furnace`. This path does not require coal but takes `scrap metal` as new input commodity. The production graph, shown in Figure 7, illustrates these additional paths. It also illustrates further possible production paths due to `liquid iron` being an output of two different steps and an input to two different steps.

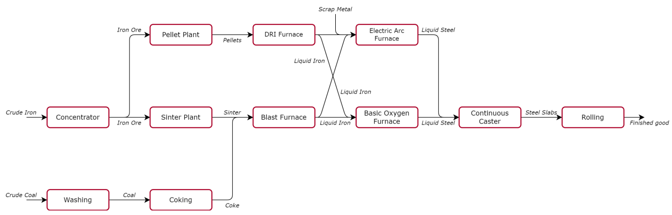


Figure 7: Production graph with multiple possible commodity flow paths, for instances I_2 and I_3. Only the `Convert` (bill of material) nodes are shown.

The physical network is the same as instance I_0, with the additional bill of materials, and `scrap metal` input commodity, spread over the existing facilities.

Instance I_3. Instance I_3 combines the production chain flexibility from I_2 with the physical network flexibility from I_1, providing the most comprehensive flexibility.

System Performance. The performance (P) of a system design $\boldsymbol{x_R}$ under a given disruption setting $\boldsymbol{x_D}$ is quantified as the total percentage of demand fulfilled for customer cities $k \in L^C$:

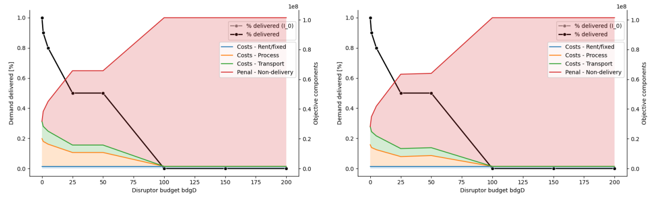
$$\mathtt{P}(\boldsymbol{x_D}, \boldsymbol{x_R}) = \sum_{k \in L^C} \sum_{p \in P} \frac{q^C_{k,p}}{dem^C_{k,p}} \tag{6.1}$$

System Resilience. The resilience score (`R-score`) of a system design is defined as the area under the performance curve (P) over varying disruptor budgets (Eq. 6.2). For a discrete set of disruptor budgets m, this is calculated using the trapezoidal rule in Eq. 6.3:

$$\begin{aligned} \mathtt{R\text{-}score}(\boldsymbol{x_R}) &= \int_0^\infty \mathtt{P} d(bdg^D) && (6.2) \\ &= \frac{1}{2} \sum_m (bdg^D_{m+1} - bdg^D_m)(\mathtt{P}_m + \mathtt{P}_{m+1}) && (6.3) \end{aligned}$$

Considering value chain flexibility. System performance (P) was analyzed as a function of the disruption budget (bdg^D). Figure 8 illustrates this relationship for simple and complex value chains without network flexibility (instances I_0 and I_2). Figure 9 shows the impact of network flexibility by varying the design budget (bdg^R) on system performance (instances I_1 and I_3). The black line represents the performance degradation with different disruption budgets (x-axis). The left y-axis displays the performance value, while the right y-axis shows the objective function value, broken down into various cost components.

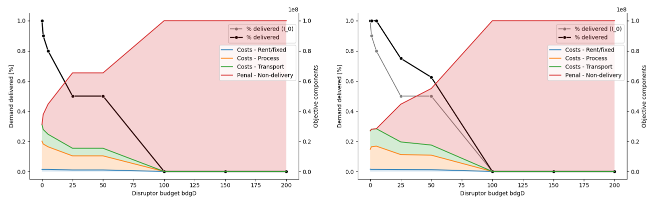
The resilience scores `R-score`, representing the area under the performance curve for different instances, are presented in Figure 10.



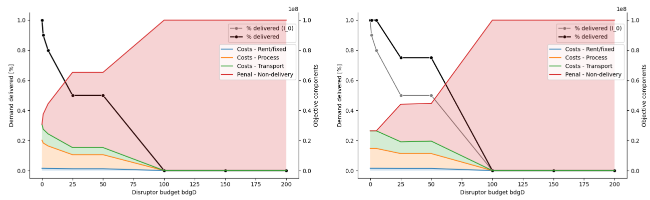
((a)) Instance I_0, $bdg^R = 0$ ((b)) Instance I_2, $bdg^R = 0$

Figure 8: Performance P and objective function components for instances I_0 and I_2.

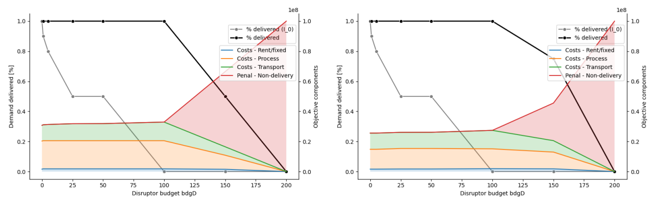
Increasing the design resource budget (bdg^R) enhances system resilience, particularly in instances with value chain flexibility. Figure 10 shows resilience scores `R-score` (y-axis) for different instances and design budgets (x-axis). A black line indicates the resilience score of the base case (I_0) without flexibility. The numbers above the bars indicate the percentage increase in resilience compared to the base case. Without value chain flexibility, resilience improves by 254% at the highest design budget ($bdg^R = 5,000,000$). With value chain flexibility, resilience increases by 30.6% at a smaller design budget ($bdg^R = 1,000,000$) and by 284% at the larger budget ($bdg^R = 5,000,000$). This demonstrates that incorporating value chain flexibility, in addition to location flexibility, significantly enhances system resilience within the same design budget.



((a)) Instance I_1, $bdg^R = 1,000,000$ ((b)) Instance I_3, $bdg^R = 1,000,000$



((c)) Instance I_1, $bdg^R = 2,000,000$ ((d)) Instance I_3, $bdg^R = 2,000,000$



((e)) Instance I_1, $bdg^R = 5,000,000$ ((f)) Instance I_3, $bdg^R = 5,000,000$

Figure 9: Effect of physical flexibility on performance P and objective function components for instances I_1 and I_3, with various design budgets bdg^R.

When examining the objective function components in Figures 8 and 9, we observe that incorporating value chain flexibility reduces production costs (shown in orange) by enabling the model to make more efficient production decisions. However, this benefit typically results in a slight increase in transportation costs (shown in green).

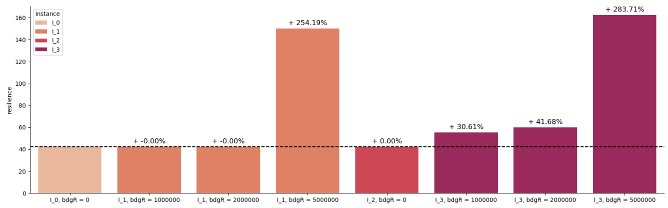


Figure 10: Resilience score `R-score` of the difference instances and design budgets bdg^R.

6.3. Assessing Resilience in a Global Pharmaceutical Supply Chain

In this section, we examine a global pharmaceutical company's supply chain and propose a method for studying climate resilience. The company produces three drugs, named `ProductA`, `ProductB`, and `ProductC`. These products are primarily sold in the US and Europe, with additional demand in Latin America, South Africa, and the Asia-Pacific region. The production process includes drug substance production, vial filling, and packaging. The products are filled into vials of varying dosages (5, 10, 15 mg, etc.) and packed into units containing 1, 2, 5, 6, or 10 vials, which are then distributed globally. Figures 11 and 12 illustrate the production steps and the involved commodities.

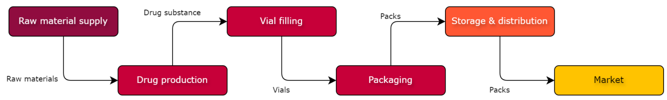


Figure 11: Diagram of production steps.

Physical network. The company's current supply chain consists of suppliers, production sites, warehouses, and demand cities. There are four production facilities in the United States, Belgium, Italy, and Ireland, each handling different production steps and products. Production resources, such as filling lines and packaging units, vary across these facilities. Figure 13 details these resources. The studied distribution network includes 30 warehouses connected to 189 demand nodes (i.e., cities), with transportation modes including truck, rail, and sea for suppliers and producers, and LTLs for customer connections.

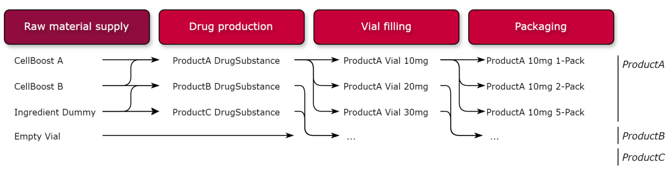


Figure 12: Overview of the commodities involved at each production step.

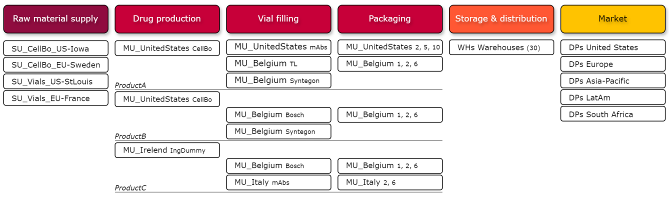


Figure 13: Distribution of existing resources across production steps. The figure illustrates that a single facility can manage multiple production steps, including various versions of those steps with different bills of materials.

Figure 14: Map showing the climate risk factors for "Drought" for the studied case.

The company plans to expand its network to meet increased demand, especially from Asia. Potential new production sites have been identified in Germany, India, Indonesia, Egypt, and Brazil, along with ten new raw material suppliers and 30 potential warehouse locations.

Value chain layout. The three main production steps, when expanded to include all sub-products, various resource configurations, and competing production processes, result in a large, interconnected complex production graph.

Instance definition. We define two instances in this experiment. A base case I_{Base} and a climate resilience case I_{Clim}. The share the same value chain and physical network, but differ in the facility disruption costs. I_{Base} assumes that facilities have equal disruption costs, unrelated to their location. The cost of disrupting facility i by impact level f is thus simply equal to a constant impact level-specific cost $c_f^{D,level}$:

$$c_{i,f}^D = c_f^{D,level} \tag{6.4}$$

Climate location-based disruption costs. In I_{Clim}, we now want to incorporate location-specific facility disruption costs, based on climate risks. Let $\theta_{i,f}^{Clim}$ be a risk factor denoting how vulnerable location i is to various climate hazards to cauase a disruption of impact level f. We can decrease the disruption costs by $1 - \theta_{i,f}^{Clim}$:

$$c_{i,f}^D = c_f^{D,level} \quad (1 - \theta_{i,f}^{Clim}) \tag{6.5}$$

We estimate $\theta_{i,f}^{Clim}$ using (slightly randomized) data from (FEMA) and RoyalHaskoningDHV (2023) on 11 different climate and geopolitcal hazards. Let $\theta_{i,r}$ be a risk score for hazard r at location i, that ranges from 0 (no risk before 2050) to 1 (high risk before 2050), and let $K_{r,f}$ relate hazards r to impact levels f, we can define the total risk factor of location i as $\theta_{i,f}^{Clim} = \sum_{r \in R} K_{r,f} \cdot \theta_{i,r}$, and thus:

$$c_{i,f}^D = c_f^{D,level} \quad (1 - \sum_{r \in R} K_{r,f} \cdot \theta_{i,r}) \tag{6.5}$$

The impact level and associated risks are presented in Appendix Appendix D. Figure 14 shows the risk scores for drought hazards.

Scenarios. We run I_{Base} to explore the effects of various Disruption and Resilience budget allocations on decision-making, system design, and commodity flow. With zero budgets, the model purely optimizes operations. A nonzero disruption budget identifies vulnerabilities, while a nonzero resilience budget focuses on building a robust supply chain. When both budgets are nonzero, the model assesses supply chain robustness under disruption scenarios.

Figures 15, 16, and 17 illustrate commodity flows with disruption and resilience budgets set to (0, 0), (100, 0), and (100, 100,000,000), respectively. New and existing facilities are distinguished by initial costs; new locations are marked in magenta, and locations to close in blue. These markings are for analytical purposes and not actual model decisions.

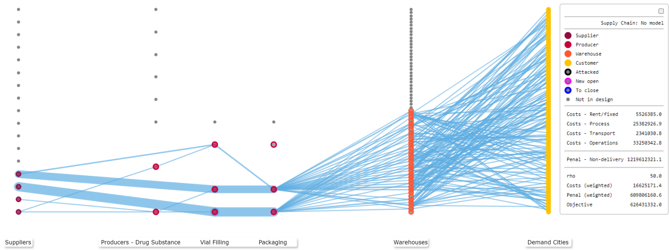


Figure 15: Operational flow diagram, $bdg^D = 0$, $bdg^R = 0$.

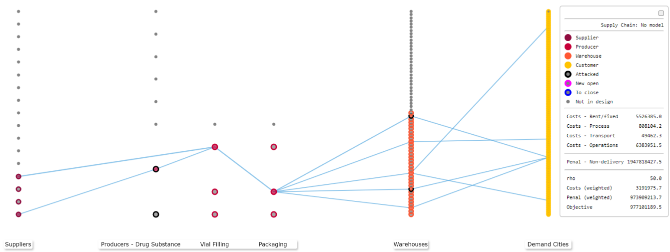


Figure 16: Operational flow diagram, $bdg^D = 100$, $bdg^R = 0$.

Effect of climate resilience. In the base case (I_{Base}), all facilities i have the same disruption cost $c_{i,f}^D$, varying only by impact level f. In the climate resilience case (I_{Clim}), we reduce the disruption cost using a location-specific risk factor $\theta_{i,f}^{Clim}$. Therefore, each facility has a unique disruption cost in the climate resilience scenario. We investigate how the model's decisions regarding facility disruptions and system design inclusion differ between the base and climate resilience cases.

Figures 18 and 19 display the frequency of facility disruptions

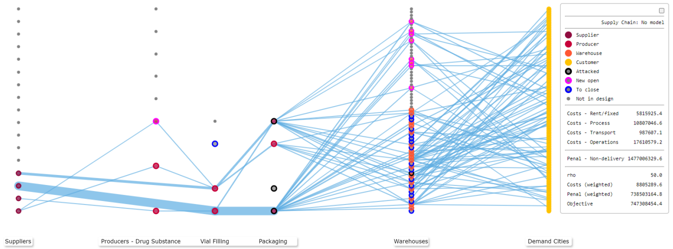


Figure 17: Operational flow diagram, $bdg^D = 100$, $bdg^R = 100,000,000$.

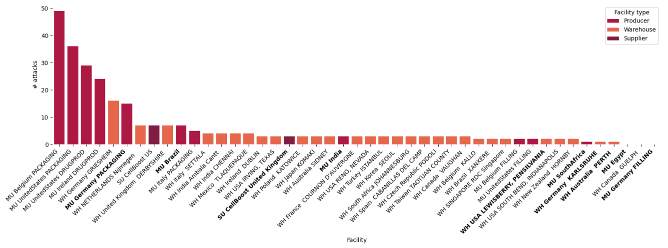


Figure 18: Frequency of disruptions per facility. Facilities in **bold** were not in the original system design (i.e., they have a nonzero initial cost).

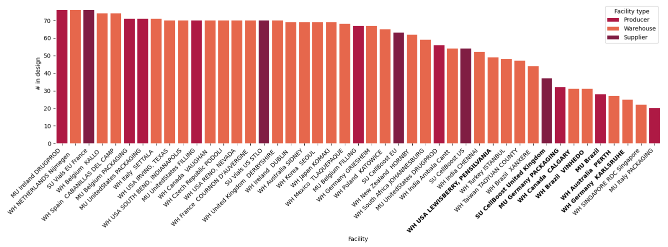


Figure 19: Frequency of facility inclusion in the design. Facilities in **bold** were not in the original system design (i.e., they have a nonzero initial cost).

and their inclusion in the system design for the base case (I_{Base}). The model primarily identifies production sites as critical vulnerabilities within the supply chain.

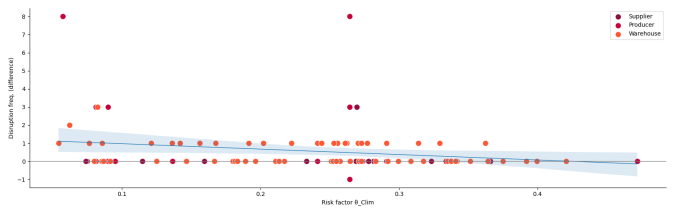


Figure 20: Relationship between a facility's climate risk factor $\theta_{i,f}^{Clim}$ and the difference in frequency of facility disruption between I_{Clim} and I_{Base}. Linear regression between both variables shows a lack of correlation.

Figures 20 and 21 illustrate the difference in disruption frequency and system design inclusion for the climate resilience case (I_{Clim}) compared to the base case (I_{Base}). The scatter plots show the risk factor $\theta_{i,f}^{Clim}$ on the x-axis and the frequency difference between I_{Clim} and I_{Base} on the y-axis. Notably, there is no correlation between increased risk hazard and disruption frequency or system design inclusion. This indicates that the facility's specific importance (e.g., single production site vs. outlying warehouse) has a greater impact on determining criticality. While this finding is case-specific, we show that our model allows for incorporating climate hazards into the resilience study of supply chains.

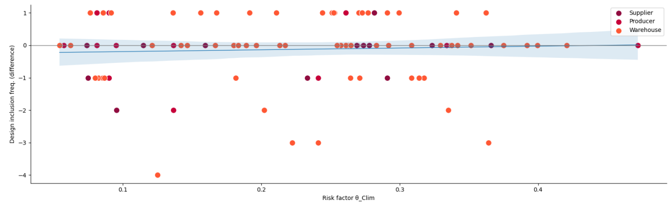


Figure 21: Relationship between a facility's climate risk factor $\theta_{i,f}^{Clim}$ and the difference in frequency of facility inclusion in design between I_{Clim} and I_{Base}. Linear regression between both variables shows a lack of correlation.

Deterministic disruptions versus simulation. In reality, disruptions have a stochastic nature that influences supply chain performance. Motivated by the stochastic nature of disruptions, in this part, we investigate whether defining disruption as an expected value differs from the deterministic case. To test our hypothesis, we compare the effectiveness of our approach to a case where disruptions are treated stochastically. We use a Monte Carlo simulation approach to identify potential disruptions within a given budget bdg^D. We generate a set of N randomly valid operational setting vectors, $\mathbf{x_A^k}$, and evaluate their impact on system performance. We compare the results with our model, which determines the optimal operational settings $\mathbf{x_A^*}$, given a system design $\mathbf{x_D}$.

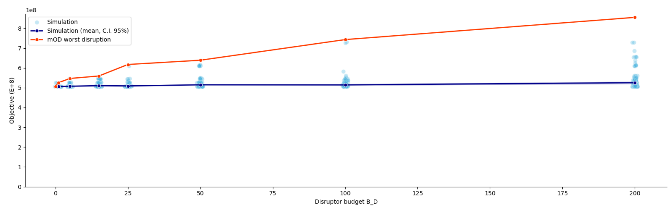


Figure 22: Objective value of optimal **mOD-MIQP** disruption $\mathbf{x_D^*}$ (orange line) versus 200 random disruptions (blue pointcloud, and mean shown), per disruption budget.

Figure 22 illustrates the results of this comparison for various disruption budgets, with the simulation approach using 200 random disruptions per bdg^D. As can be seen in this figure, the simulation approach requires significantly more iterations to determine the combinations that can lead to system failure. Due to the correlation between disruptions, the simulation model requires a significantly larger number of draws to identify the entire scenario. The average (disrupted) objective value lays around 25% lower for $bdg^D = 50$ and 40% lower for $bdg^D = 200$. As a result, relying on simulation to determine the average or most common disruptions may significantly deviate from the most severe cases, which can be determined by the mode.

7. Conclusion

This paper has presented **mODR**, a model for resilient supply chain design and operation. The tri-level optimization model models disruptions such as supplier failures and production shortfalls through capacity reduction. It enables contingent rerouting and strategic design of supply chains for increased capacity and multi-sourcing. The model is decomposed into an disruptor sub-problem and a master problem, which are solved iterative.

The computational performance of our approach proved efficient in solving a diverse set of realistically sized cases within reasonable time frames. Our analysis demonstrated that incorporating value chain flexibility, in addition to physical network flexibility, enhances the system's resilience, especially with increasing design budgets. Furthermore, applying our model to a real-life pharmaceutical supply chain highlighted its capability to integrate climate hazards into resilience modeling, underscoring its practical utility. Finally, the comparison between deterministic optimization and simulation approaches underscores the efficiency of the deterministic method in identifying optimal disruptions.

We conclude that the **mODR** model is a valuable tool for supply chain managers in designing resilient supply chains. It effectively meets the goals of developing a computationally tractable model capable of addressing facility capacity disruptions and implementing both contingent and strategic resilience measures. Our findings demonstrate that supply chain resilience is rooted not only in the physical network's flexibility but also in the adaptability of the value and production chains.

However, the model has limitations, including its reliance on a continuous LP for the operator problem. Future research should explore non-LP operator problems. Further research should incorporate a temporal component into the spatial supply chain model to account for time-dependent factors like production processes, transportation delays, and perishable goods, enhancing the model's ability to assess resilience at both strategic and tactical levels. Additionally, further studies should develop more robust methods for determining disruption budgets and costs based on theoretical risk analysis, as these significantly impact the model's effectiveness in identifying critical supply chain vulnerabilities.

References

Alderson, D.L., Brown, G.G., Carlyle, W.M., Wood, R.K., 2011. Solving defender-attacker-defender models for infrastructure defense. Center for Infrastructure Defense, Operations Research Department, Naval Postgraduate School doi:10.1287/ics.2011.0047.

Dempe, S., Zemkoho, A. (Eds.), 2020. Bilevel Optimization: Advances and Next Challenges. volume 161 of *Springer Optimization and Its Applications*. Springer, Cham.

Fattahi, M., Govindan, K., Keyvanshokooh, E., 2018. A multi-stage stochastic program for supply chain network redesign problem with price-dependent uncertain demands. Computers & Operations Research 100, 314–332. URL: https://www.sciencedirect.com/science/article/pii/S0305054817303167, doi:https://doi.org/10.1016/j.cor.2017.12.016.

(FEMA), F.E.M.A., 2023. National risk index map. URL: https://hazards.fema.gov/nri/map. accessed: 2024-06-18.

Folke, C., 2006. Resilience: The emergence of a perspective for social–ecological systems analyses. Global Environmental Change 16, 253–267. URL: https://www.sciencedirect.com/science/article/pii/S0959378006000379, doi:https://doi.org/10.1016/j.gloenvcha.2006.04.002. resilience, Vulnerability, and Adaptation: A Cross-Cutting Theme of the International Human Dimensions Programme on Global Environmental Change.

Ghorbani-Renani, N., González, A.D., Barker, K., 2021. A decomposition approach for solving tri-level defender-attacker-defender problems. Computers & Industrial Engineering 153, 107085.

Govindan, K., Fattahi, M., Keyvanshokooh, E., 2017. Supply chain network design under uncertainty: A comprehensive review and future research directions. European Journal of Operational Research URL: www.elsevier.com/locate/ejor. production, Manufacturing and Logistics.

Hamed Soleimani, M.S.E., Kannan, G., 2014. Incorporating risk measures in closed-loop supply chain network design. International Journal of Production Research 52, 1843–1867. URL: https://doi.org/10.1080/00207543.2013.849823, doi:10.1080/00207543.2013.849823.

Harrison, A., Godsell, J., 2003. Responsive supply chains: An exploratory study of performance management .

Ho, W., Zheng, T., Yildiz, H., Talluri, S., 2015. Supply chain risk management: A literature review. International Journal of Production Research 53. doi:10.1080/00207543.2015.1030467. authors' affiliations: University of Melbourne, Wayne State University, Michigan State University.

Keyvanshokooh, E., Ryan, S.M., Kabir, E., 2016. Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated benders decomposition. European Journal of Operational Research 249, 76–92. URL: https://www.sciencedirect.com/science/article/pii/S0377221715007833, doi:https://doi.org/10.1016/j.ejor.2015.08.028.

Klibi, W., Martel, A., Guitouni, A., 2010. The design of robust value-creating supply chain networks: A critical review. European Journal of Operational Research 203, 283–293. URL: https://www.sciencedirect.com/science/article/pii/S0377221709004792, doi:https://doi.org/10.1016/j.ejor.2009.06.011.

Li, X., Kizito, R., Paula, T.I., 2018. An agent-based simulation framework for supply chain disruptions and facility fortification , 821–832doi:10.1109/WSC.2018.8632475.

Liu, Y., Guo, B., 2014. A lexicographic approach to postdisaster relief logistics planning considering fill rates and costs under uncertainty. Volume 2014 URL: https://doi.org/10.1155/2014/939853, doi:10.1155/2014/939853. open Access, Academic Editor: Lu Zhen.

Rosenhead, J., Elton, M., Gupta, S.K., 1972. Robustness and optimality as criteria for strategic decisions. Journal of the Operational Research Society 23, 20.

RoyalHaskoningDHV, 2023. Climate resilience app. URL: https://www.royalhaskoningdhv.com/en/challenges/resilience/climate-resilience/climate-hazard-and-risk-insights-and-awareness. accessed: 2024-06-18.

Snyder, L.V., Atan, Z., Peng, P., Rong, Y., Schmitt, A.J., Sinsoysal, B., 2014. Or/ms models for supply chain disruptions: A review .

Sweney, M., 2021. Global shortage in computer chips 'reaches crisis point'. The Guardian Available: https://www.theguardian.com/business/2021/mar/21/global-shortage-in-computer-chips-reaches-crisis-point, Accessed: 2024-08-31.

Wang, D., Peng, J., Yang, H., Cheng, T., Yang, Y., 2023. Distributionally robust location-allocation with demand and facility disruption uncertainties in emergency logistics. Computers Industrial Engineering 184, 109617. URL: https://www.sciencedirect.com/science/article/pii/S0360835223006411, doi:https://doi.org/10.1016/j.cie.2023.109617.
Wieland, A., Durach, C.F., 2021. Two perspectives on supply chain resilience. Journal of Business Logistics 42, 233–249. doi:10.1111/jbl.12271. [Article updated on March 17, 2021 after first online publication: reference to Rao et al. (2021) was corrected to Iyengar et al. (2021)].
Wieland, A., Wallenburg, C.M., 2011. Supply-Chain-Management in stürmischen Zeiten. Universitätsverlag der TU, Berlin. URL: https://www.gbv.de.
Xu, J., Zhuang, J., Liu, Z., 2016. Modeling and mitigating the effects of supply chain disruption in a defender–attacker game. Annals of Operations Research 236, 255–270. doi:10.1007/s10479-015-1810-z.
Yuan, W., Zhao, L., Zeng, B., 2014. Optimal power grid protection through a defender–attacker–defender model. Reliability Engineering & System Safety 121, 83–89. doi:10.1016/j.ress.2013.08.003.
Zhen, L., Zhuge, D., Lei, J., 2016. Supply chain optimization in context of production flow network. Journal of Systems Science and Systems Engineering 25, 351–369. URL: https://doi.org/10.1007/s11518-016-5304-6, doi:10.1007/s11518-016-5304-6.

Appendix A. Table of Notations

Table A.9: Notations for Value Chain

Sets	**Description**
$p \in P$	Commodities, including raw materials and products
$b \in B$	Bills of Materials (BoM), specifying conversion processes
Parameters	
$g^{in}_{b,p}$	Input quantity of commodity p per BoM b
$g^{out}_{b,p}$	Output quantity of commodity p per BoM b

Table A.10: Notations for Physical Network

Sets	**Description**
$i, j \in L$	Physical locations in the supply chain
$h \in L^S \subseteq L$	Supplier locations
$i \in L^P \subseteq L$	Producer locations
$j \in L^W \subseteq L$	Warehouse locations
$k \in L^C \subseteq L$	Customer locations
$(i, j) \in A$	Links between locations i and j
$p \in P(i, j)$	Commodities flowing on link (i, j)
$m \in M(i, j)$	Transport modes available on link (i, j)
Parameters	
$cap^S_{h,p}$	Supply capacity of p by supplier h
$cap^P_{i,b}$	Production capacity for BoM b at producer i
$cap^W_{j,p}$	Storage capacity of p at warehouse j
$dem^C_{k,p}$	Demand for p by customer k
c^{init}_i	Initial investment cost for location i
c^{fix}_i	Fixed operational cost for location i
$c^{pr}_{i,p}$	Processing cost for p at location i
c^{trip}_m	Fixed cost to start a trip using mode m
c^{dist}_m	Variable cost per unit distance using mode m
$d_{i,j}$	Distance between locations i and j
ls_m	Load size capacity of mode m
lc_p	Load unit conversion factor for p
$c^{tr}_{i,j,m}$	Transport cost between i and j using mode m

Table A.11: Notations - Operational setting parameters

Operational setting	
bdg^D	Total available attack budget
$c^{att,loc}_i$	Attack cost of fully disabling facility i
$c^{att,pr,S}_{h,p,f}$	Attack cost of partially disabling the supply of commodity p at supplier h by level f
$c^{att,pr,P}_{i,b,f}$	Attack cost of partially disabling the production of bill of material b at producer i by level f
u_f	Impact (% disabled) of disruption level f

Table A.12: Notations - System design parameters

System design	
bdg^R	Total available system design / defence budget
$c^{init,S}_h$	Initial investment cost of using supplier h
$c^{init,P}_i$	Initial investment cost of using producer i
$c^{init,W}_j$	Initial investment cost of using warehouse j

Table A.13: Notations - Operator decision variables $\boldsymbol{x_O}$

Operations x_O	
$q^S_{h,p}$	Amount of commodity p supplied from supplier h
$q^P_{i,b}$	Amount of bill of material b produced at producer i
$q^W_{j,p}$	Amount of commodity p stored at warehouse j
$q^C_{k,p}$	Amount of commodity p delivered to customer k
$\bar{q}^C_{k,p}$	Amount of commodity p *not* delivered to customer k (difference from demand)
$y_{i,j,p}$	Amount of commodity p between locations i and j
$z_{i,j,m}$	Number of trips between locations i and j using mode m

Table A.14: Notations - Disruption decision variables $\boldsymbol{x_D}$

Disruption x_D	
ϕ^S_h	Binary variable, 1 if supplier location h is fully disabled
ϕ^P_i	Binary variable, 1 if producer location i is fully disabled
ϕ^W_j	Binary variable, 1 if warehouse location j is fully disabled
$\psi^S_{h,p,f}$	Binary variable, 1 if the supply of commodity p from supplier h is disrupted by level f
$\psi^P_{i,b,f}$	Binary variable, 1 if the production of bill of material b at producer i is disrupted by level f

Table A.15: Notations - System design decision variables $\boldsymbol{x_R}$

System design x_R	
x^S_h	Binary variable, 1 if supplier location h is used
x^P_i	Binary variable, 1 if producer location i is used
x^W_j	Binary variable, 1 if warehouse location j is used

Appendix B. Full mOD-MIQP definition

Tables B.16 and B.17 show the dual variables and constraints of **mO**. We define **mOD-MIQP** $(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) \rightarrow \boldsymbol{x_\Omega}^*, \boldsymbol{x_D}^*$ as follows:

[mOD-MIQP]

$$\max_{\boldsymbol{x_\Omega}, \boldsymbol{x_D}} \quad \Delta^{oper,dual} \quad (\hat{\boldsymbol{x}}_{\boldsymbol{R}}) \tag{B.1}$$

$$\text{s.t.} \quad \upsilon^S_{h,p} + \nu^S_{h,p} + \sum_{f \in F} \gamma^S_{h,p,f} - \alpha^S_{h,p} \leq c^{pr,S}_{h,p} \cdot \rho^c \qquad \forall h \in L^S, \quad \forall p \in P \tag{B.2}$$

$$\upsilon^P_{i,b} + \nu^P_{i,b} + \sum_{f \in F} \gamma^P_{i,b,f} - \sum_{p \in P} (g^{in}_{b,p} \cdot b^{in}_{i,p} - g^{out}_{b,p} \cdot b^{out}_{i,p}) \leq c^{pr,P}_{i,b} \cdot \rho^c \qquad \forall i \in L^P, \quad \forall b \in B \tag{B.3}$$

$$\upsilon^W_{j,p} + \nu^W_{j,p} - \alpha^{W,in}_{j,p} - \alpha^{W,out}_{j,p} \leq c^{pr,W}_{j,p} \cdot \rho^c \qquad \forall j \in L^W, \quad \forall p \in P \tag{B.4}$$

$$\delta^C_{k,p} - \alpha^C_{k,p} \leq 0 \qquad \forall k \in L^C, \quad \forall p \in P \tag{B.5}$$

$$\delta^C_{k,p} \leq c^{rv,C}_{k,p} \cdot \rho^r \qquad \forall k \in L^C, \quad \forall p \in P \tag{B.6}$$

$$\alpha^S_{i,p} + \alpha^{W,in}_{j,p} + \alpha^{W,out}_{i,p} + \alpha^C_{j,p} + \beta^{in}_{i,p} + \beta^{out}_{i,p} + lc_p \cdot \mu_{i,j} \leq 0 \qquad \forall (i,j) \in A, \quad \forall p \in P(i,j) \tag{B.7}$$

$$-ls_m \cdot \mu_{i,j} \leq c^{tr}_{i,j,m} \cdot \rho^c \qquad \forall (i,j) \in A, \quad \forall m \in M(i,j) \tag{B.8}$$

$$\Gamma^D \leq bdg^D \tag{4.24}$$

$$\boldsymbol{\upsilon}, \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\mu} \leq \boldsymbol{0} \tag{B.9}$$

$$\boldsymbol{\alpha}, \boldsymbol{\delta} \in \mathbb{R} \tag{B.10}$$

$$\boldsymbol{\phi}, \boldsymbol{\psi} \in \{0,1\} \qquad \text{(4.19) - (4.23)}$$

The objective function (B.1) of **mOD-MIQP** is defined as follows. Note that it reuses the fixed cost term $\Gamma^{fixed}(\hat{\boldsymbol{x}}_{\boldsymbol{R}})$ from the objective function of **mODR** (4.29).

$$\Delta^{oper,dual}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) = \Gamma^{fixed}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) + \Delta^{cap,S}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) + \Delta^{cap,P}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) + \Delta^{cap,W}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) + \Delta^{dem} \tag{B.1}$$

Where the individual objective terms are calculated as follows:

$$\begin{aligned}
\Delta^{cap,S}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) &= \sum_{h \in L^S} \sum_{p \in P} cap^S_{h,p} \cdot (\hat{x^S_h} \cdot \upsilon^S_{h,p} + (1 - \phi^S_h) \cdot \nu^S_{h,p} \\
&\quad + \sum_{f \in F} (1 - u_f \cdot \psi^S_{h,p,f}) \cdot \gamma^S_{h,p,f}) \\
\Delta^{cap,P}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) &= \sum_{i \in L^P} \sum_{b \in B} cap^P_{i,b} \cdot (\hat{x^P_i} \cdot \upsilon^P_{i,b} + (1 - \phi^P_i) \cdot \nu^P_{i,b} \\
&\quad + \sum_{f \in F} (1 - u_f \cdot \psi^P_{i,b,f}) \cdot \gamma^P_{i,b,f}) \\
\Delta^{cap,W}(\hat{\boldsymbol{x}}_{\boldsymbol{R}}) &= \sum_{j \in L^W} \sum_{p \in P} cap^W_{j,p} \cdot (\hat{x^W_j} \cdot \upsilon^W_{j,p} + (1 - \phi^W_j) \cdot \nu^W_{j,p}) \\
\Delta^{dem} &= \sum_{k \in L^C} \sum_{p \in P} dem^C_{k,p} \cdot \delta^C_{k,p}
\end{aligned}$$

mOD-MIQP is nonlinear due to its objective function, where dual operator variables are multiplied by disruption variables. Specifically, the nonlinearity arises from the terms $(1 - \phi_i) \cdot \nu_{i,p}$ and $(1 - u_f \cdot \psi_{i,p,f}) \cdot \gamma_{i,p,f}$.

Table B.16: **mO-Dual** dual constraints definitions.

Variables **mO**	Sets	Constraints **mO-Dual**
$q^S_{h,p}$	$\forall h \in L^S, \quad \forall p \in P$	B.2
$q^P_{i,b}$	$\forall i \in L^P, \quad \forall b \in B$	B.3
$q^W_{j,p}$	$\forall j \in L^W, \quad \forall p \in P$	B.4
$q^C_{k,p}$	$\forall k \in L^C, \quad \forall p \in P$	B.5
$\bar{q}^C_{k,p}$	$\forall k \in L^C, \quad \forall p \in P$	B.6
$y_{i,j,p}$	$\forall (i,j) \in A, \quad \forall p \in P(i,j)$	B.7
$z_{i,j,m}$	$\forall (i,j) \in A, \quad \forall p \in M(i,j)$	B.8

Table B.17: **mO-Dual** dual variable definitions.

Constraints **mO**	Description	Variables **mO-Dual**
4.2	Supplier capacity	$\upsilon^S_{h,p}, \quad \nu^S_{h,p}, \quad \gamma^S_{h,p,f}$
4.3	Producer capacity	$\upsilon^P_{i,b}, \quad \nu^P_{i,b}, \quad \gamma^P_{i,b,f}$
4.4	Warehouse capacity	$\upsilon^W_{j,p}, \quad \nu^W_{j,p}$
4.8	Supplier flow balance outgoing	$\alpha^S_{h,p}$
4.9	Warehouse flow balance	$\alpha^{W,in}_{j,p}, \quad \alpha^{W,out}_{j,p}$
4.10	Customer flow balance incoming	$\alpha^C_{k,p}$
4.15	BoM balance incoming flows	$\beta^{P,in}_{i,p}$
4.16	BoM balance outgoing flows	$\beta^{P,out}_{i,p}$
4.12	Demand and delivery balance	$\delta^C_{k,p}$
4.17	Transport load	$\mu_{i,j}$

Appendix C. Computational results tables

Table C.18: Computational performance - Budget combinations

	None	None Mild	None Moderate	None Severe	None Catastrophic	Minimal None	Minimal Mild	Minimal Moderate	Minimal Severe	Minimal Catastrophic	Moderate None	Moderate Mild	Moderate Moderate
time mean	0.05	0.58	0.79	0.68	0.62	0.07	2.53	3.66	3.96	4.12	0.19	122.63	407.56
time min	0.00	0.03	0.03	0.04	0.03	0.01	0.18	0.15	0.14	0.24	0.01	0.24	0.30
time max	0.17	2.06	3.35	2.50	2.15	0.26	9.39	16.78	15.65	14.35	0.63	1023.97	4290.81
gap mean	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gap min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gap max	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	Moderate Severe	Moderate Catastrophic	Extensive None	Extensive Mild	Extensive Moderate	Extensive Severe	Extensive Catastrophic	Global None	Global Mild	Global Moderate	Global Severe	Global Catastrophic
time mean	57.66	17.90	0.61	512.94	1278.21	712.84	1800.11	0.78	863.99	1679.86	1432.04	2408.14
time min	0.26	0.24	0.01	0.10	0.12	0.19	2.39	0.01	0.08	0.12	0.12	1.42
time max	382.06	161.47	2.44	5400.00	5400.00	5400.00	5400.00	2.73	5400.00	5400.00	5400.00	5400.00
gap mean	0.00	0.00	0.00	0.89	5.50	2.95	37.47	0.00	0.91	5.53	9.06	38.59
gap min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gap max	0.00	0.00	0.00	10.73	50.96	35.42	174.23	0.00	10.98	66.13	108.72	221.10

Table C.19: Computational results for `VC-Simple`

	Network size	`Small`					`Medium`					`Large`				
	Γ^R	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**
Γ^D	*Results*															
None	**Solve time [s]**	0.00	0.01	0.01	0.02	0.01	0.01	0.01	0.10	0.42	0.12	0.04	0.06	0.39	2.06	2.62
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.03	0.18	0.24	0.10	0.08	0.13	1.12	3.78	5.59	0.60	0.53	2.20	71.39	3.88	4.43
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00
Moderate	**Solve time [s]**	0.03	0.15	0.30	0.12	0.12	0.12	0.59	41.98	261.42	0.81	0.47	2.25	392.79	3.90	4.44
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Severe	**Solve time [s]**	0.04	0.14	0.26	0.19	0.12	0.12	0.63	42.23	263.82	0.73	0.46	2.25	382.06	4.05	4.70
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Catastrophic	**Solve time [s]**	0.03	0.24	0.24	2.49	5.34	0.13	1.63	1.49	**5400.00**	1787.82	0.53	5.34	5.88	**5400.00**	5400.00
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**173.64**	0.00	0.00	0.00	0.00	**174.23**	0.58
	Delivered [%]	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	100.00	0.00	0.00	0.00	100.00	100.00

Table C.20: Computational results for `VC-LinSingl`

	Network size	`Small`					`Medium`					`Large`				
	Γ^R	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**
Γ^D	*Results*															
None	**Solve time [s]**	0.01	0.01	0.02	0.02	0.02	0.01	0.02	0.22	0.32	0.14	0.06	0.07	0.29	1.43	1.77
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.06	0.53	0.77	0.57	0.69	0.21	1.11	4.41	12.80	9.48	0.73	2.96	1023.97	586.64	387.14
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00	90.00	90.00	90.00	100.00	100.00
Moderate	**Solve time [s]**	0.06	0.41	0.62	2.60	0.59	0.29	1.15	5.86	84.93	21.81	0.61	3.00	57.81	**5400.00**	5190.04
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**50.96**	0.21
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Severe	**Solve time [s]**	0.05	0.24	0.55	1.38	0.81	0.20	0.88	5.66	11.84	11.38	1.62	4.60	107.02	2132.22	5283.48
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05
	Delivered [%]	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00	50.00	50.00	50.00	100.00	100.00
Catastrophic	**Solve time [s]**	0.08	0.28	0.73	3.70	4.35	0.27	1.26	6.04	2299.32	**5400.00**	0.68	3.10	7.71	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**102.87**	0.00	0.00	0.00	**101.71**	**103.23**
	Delivered [%]	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00	100.00

Table C.21: Computational results for `VC-LinMulti`

	Network size	`Small`					`Medium`					`Large`				
	Γ^R	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**
Γ^D	*Results*															
None	**Solve time [s]**	0.01	0.01	0.01	0.01	0.01	0.02	0.05	0.11	0.17	1.22	0.09	0.12	0.63	2.44	2.73
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.10	0.47	0.49	0.42	0.50	0.32	1.42	6.73	62.16	2885.99	2.37	10.10	594.24	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**10.73**	**10.98**
	Delivered [%]	96.67	96.67	96.67	96.67	96.67	96.67	96.67	96.67	96.67	100.00	96.67	96.67	96.67	100.00	100.00
Moderate	**Solve time [s]**	0.22	0.49	0.47	0.75	0.49	0.42	1.59	65.08	3977.50	5357.82	2.24	16.78	4290.81	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**15.08**	**66.13**
	Delivered [%]	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	83.33	100.00	80.00	80.00	86.67	93.33	100.00
Severe	**Solve time [s]**	0.22	0.58	0.61	0.80	0.54	0.39	2.65	11.62	568.13	1077.06	1.24	15.65	92.26	**5400.00**	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**35.42**	**108.72**
	Delivered [%]	66.67	66.67	66.67	66.67	66.67	66.67	66.67	66.67	66.67	100.00	66.67	66.67	66.67	100.00	100.00
Catastrophic	**Solve time [s]**	0.31	0.93	1.47	2.39	1.42	0.54	5.25	5.13	195.72	**5400.00**	1.36	14.35	161.47	2857.83	**5400.00**
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**35.29**	0.00	0.00	0.00	0.00	**221.10**
	Delivered [%]	33.33	33.33	33.33	33.33	33.33	33.33	33.33	33.33	33.33	100.00	33.33	33.33	33.33	33.33	100.00

Table C.22: Computational results for `VC-Complex`

	Network size	`Small`					`Medium`					`Large`				
	Γ^R	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**	**None**	**Minimal**	**Moderate**	**Extensive**	**Global**
Γ^D	*Results*															
None	**Solve time [s]**	0.02	0.01	0.03	0.02	0.03	0.09	0.26	0.10	0.08	0.15	0.17	0.17	0.34	0.34	0.48
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mild	**Solve time [s]**	0.55	2.27	2.04	2.48	2.70	1.24	4.10	4.30	3.62	3.58	2.06	9.39	24.80	77.03	1672.71
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00
Moderate	**Solve time [s]**	0.67	2.28	2.36	2.44	2.32	0.99	4.39	7.66	4.41	4.32	3.35	10.88	24.99	200.51	4175.57
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Severe	**Solve time [s]**	0.44	2.16	2.19	1.96	2.05	0.88	6.42	6.60	3.62	3.59	2.50	11.37	40.83	166.07	5400.00
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Catastrophic	**Solve time [s]**	0.55	3.82	3.16	3.19	3.22	0.80	3.51	3.63	3.60	3.54	2.15	9.74	17.85	33.09	92.05
	Gap [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Delivered [%]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Appendix D. Case study tables

.

Table D.23: Climate hazard and impact level definitions

Impact level f	**Impact u_f [%]**	**Level bias k_f**	**Risks $K_{r,f}$**	**Heat**	**Cold**	**Rain**	**Snow**	**TropicalStorm**	**Wildfire**	**AirQuality**	**Flood**	**Drought**	**Earthquake**	**HumanConflict**
`MINOR`	10	1		0.020	0.020	0.020	0.020	0.005	0.005	0.010	0.005	0.020	0.005	0.005
`HEAVY`	20	4		0.020	0.020	0.020	0.020	0.010	0.010	0.005	0.010	0.020	0.010	0.010
`MAJOR`	50	25		0.010	0.010	0.010	0.010	0.050	0.050	0.005	0.050	0.010	0.050	0.050
`FATAL`	100	100		0.005	0.005	0.005	0.005	0.050	0.050	0.001	0.050	0.010	0.050	0.050

B

Network graph construction

The Supply Chain Decision Network, as described above, is the graph of all possible locations and links upon which decisions can be made by the optimization model. This graph, denoted $G^{SC}(L, A)$, is constructed from the input data. First, after loading in the data on the components described above, a so-called Production Graph G^{PROD} is created from the BillOfMaterials and Commodities data. This is an abstract graph of the different process steps and how commodity flows may connect these steps (akin to the system's value chain), and is constructed using procedure 3. The definition of the Production Graph is essential to the allowed topology of the Supply Chain Decision Network. Note that in the G^{PROD} resulting from procedure 3, CONVER nodes can be connected directly to `STORE` and `SINK` nodes, which would allow for direct connections between producers and customers.

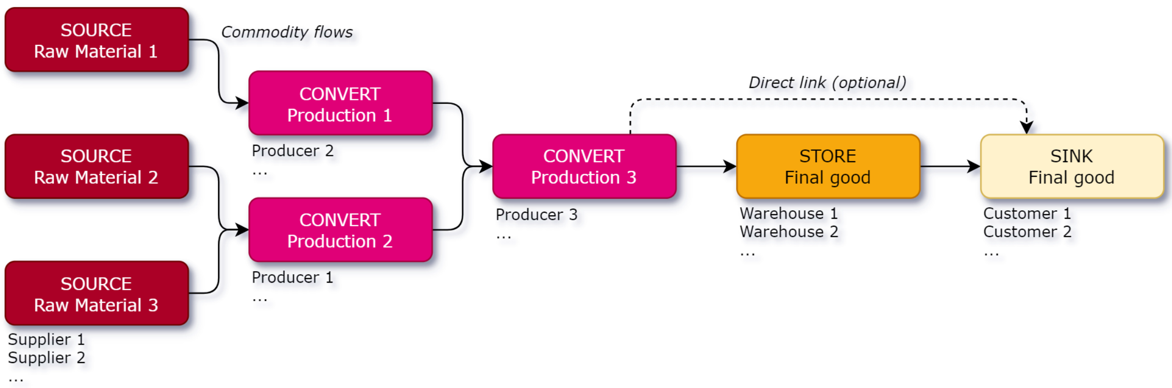


Figure B.1: Example Production Graph G^{PROD} (with some locations listed).

ID	name	unit	unit_eq
RAW1	Raw Material 1	kg	1
RAW2	Raw Material 2	kg	1
RAW3	Raw Material 3	kg	1
INT1	Intermediate 1	each	10
INT2	Intermediate 2	each	10
FINAL	Final Product	each	50

ID	name	comm_in_ID	comm_in_name	comm_out_ID	comm_out_name	unit	amount_in	amount_out
Prod1BOM	BOM Production 1			INT1	Intermediate 1	each		1
Prod1BOM	BOM Production 1	RAW1	Raw Material 1			kg	10	
Prod2BOM	BOM Production 2			INT2	Intermediate 2	each		1
Prod2BOM	BOM Production 2	RAW2	Raw Material 2			kg	10	
Prod2BOM	BOM Production 2	RAW3	Raw Material 3			kg	10	
Prod3BOM	BOM Production 3			FINAL	Final Product	each		1
Prod3BOM	BOM Production 3	INT1	Intermediate 1			each	2	
Prod3BOM	BOM Production 3	INT2	Intermediate 2			each	2	

Figure B.2: Example Production Graph input for G^{PROD}.

Algorithm 3 | Production Graph G^{PROD} construction

Input: Input data on BillOfMaterial processes and Commodities
Output: Production graph G^{PROD}

Initialize an empty graph G^{PROD}
foreach *BillOfMaterial process* b **do**
    Add a `CONVERT` node for b to G^{PROD}
end
foreach *Commodity* p **do**
    Add a `SOURCE` node for p to G^{PROD}
    Add a `STORE` node for p to G^{PROD}
    Add a `SINK` node for p to G^{PROD}
end
foreach *Commodity* p **do**
    foreach *pair of nodes* N_1 *and* N_2 *in* G^{PROD} **do**
        // A SOURCE node associated with a Commodity p has an output of p. Its STORE node has both an input and output, and its SINK node has only an input.
        // A CONVERT node has an input of Commodity p if p is a required input of the associated BillOfMaterial b, or an output of p if it is an output product b
        if N_1 *has an output of* p ***and*** N_2 *has an input of* p **then**
            // There is a potential flow of p between N_1 and N_2
            if ***not*** *edge* (N_1, N_2) *exists in* G^{PROD} **then**
                Add edge (N_1, N_2) to G^{PROD}
            end
            Add p to the $commodities$ attribute of edge (N_1, N_2)
        end
    end
end

If customers can only be connected to warehouses, one should remove the edges between CONVERT and `SINK` nodes; `SINK` nodes will then only be connected to `STORE` nodes. Alternatively, `STORE` nodes can be removed altogether if dealing with a purely production system instead of a production-distribution system. More complex layouts can be devised at this step as well. We can define multiple types of `STORE` nodes and their allowed links to simulate systems with multiple storage steps (e.g., for international flows) or systems with **trans-shipments**. The Supply Chain Decision Graph $G^{SC}(L, A)$ can be constructed by combining the Production Graph G^{PROD} and the location data using procedure 4.

Algorithm 4 | Supply Chain Decision Graph $G^{SC}(L, A)$ construction

Input: Production graph G^{PROD}; Input data on Locations L
Output: Supply Chain Decision Graph $G^{SC}(L, A)$

foreach *BillOfMaterial* b **do**
  foreach *Producer* $i \in L^P$ **do**
    if *Capacity of* b *at* $i \geq 0$ **then**
      Add i to $locations$ of CONVERT node for b
    end
  end
end
foreach *Commodity* p **do**
  foreach *Supplier* $h \in L^S$ **do**
    if *Supply capacity of* p *at* $h \geq 0$ **then**
      Add h to $locations$ of SOURCE node for p
    end
  end
  // Do the same for the Warehouses and Customers and the STORE and SINK nodes, respectively.
end
Initialize an empty graph G^{SC}
foreach *Location* $i \in L$ **do**
  Add a node i to G^{SC}
end
foreach *Edge* (N_1, N_2) *in* G^{PROD} **do**
  Add edges (i, j) to G^{SC} for each location i in N_1 and each location j in N_2
  Update the possible flows on link (i, j) with the defined Commodities on (N_1, N_2)
end

Figure B.1 depicts an example of a Production Graph. Figure B.3 provides an overview of the supply chain model, depicting the location layers and the interconnecting links. The diagram illustrates a specific supply chain solution (colored nodes and darkened links) atop all potential supply chain configurations which form the Supply Chain Decision Graph $G^{SC}(L, A)$ (grayed-out nodes and links).

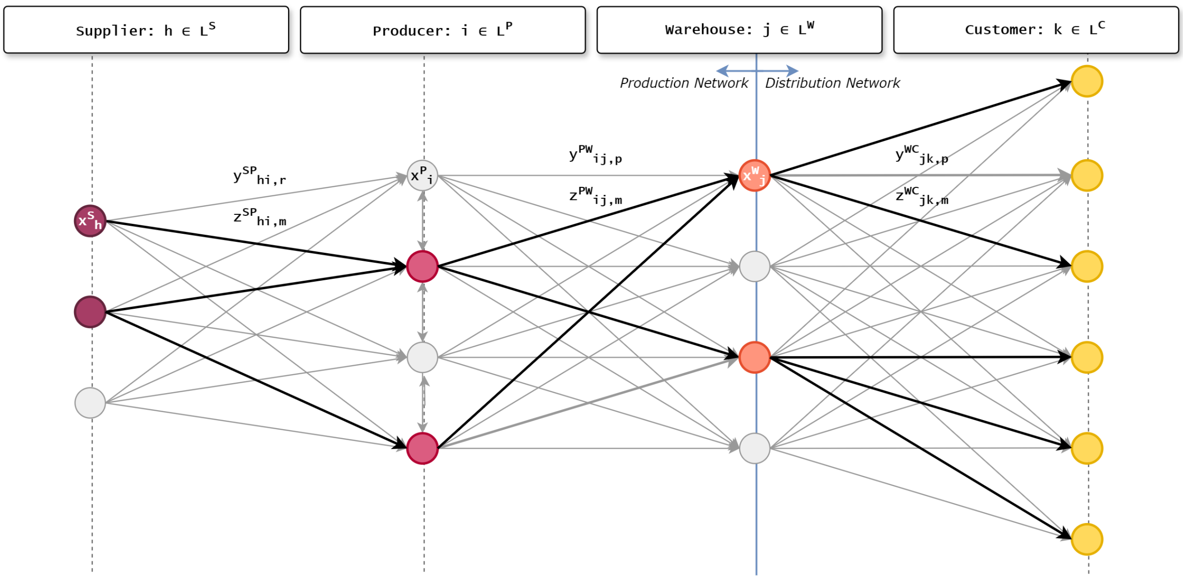


Figure B.3: Overview of the Supply Chain Decision Graph $G^{SC}(L, A)$, our supply chain model.