A Hybrid Simulation Methodology for Identifying and Mitigating Supply Chain Disruptions

Abhisekh Rana[1], Raj Patel[1], Aman Goswami[2], Sean Luke[1], Alok Baveja[2], Carlotta Domeniconi[1], Benjamin Melamed[2], Fred Roberts[2], Weiwei Chen[2], Andrew Crooks[3], Vladimir Menkov[1], Viswanath Narayan[4], James Jones[1], and Hamdi Kavak*[1]

[1]George Mason University

[2]Rutgers University

[3]University at Buffalo

[4]Retired pharmaceutical executive

Abstract

The recent COVID pandemic has demonstrated how disruptions to supply chains can negatively impact the world economy. The interconnectedness of global supply chains means that the effects of these disruptions can propagate far beyond their origination point. In this paper, we propose a simulation methodology, dubbed Hybrid Integrated Supply-Chain Simulation (HISS), to identify and mitigate potential disruptions in supply chains. We illustrate the HISS methodology through a generic pharmaceutical supply chain, created using MASON's novel hybrid modeling capabilities of the ECJ stochastic optimization toolkit. The model captures the sourcing, outsourcing, production, packaging, and distribution processes of a pharmaceutical company. We provide a detailed classification of disruptions to this supply chain from malicious actors and overlay them onto the supply chain map. The simulation model is used to study the timing, impact, and scope of these disruptions. The simulation is further extended to modeling pre- and post-disruption mitigation strategies and assessing their efficacy. Extensive optimization allowed us to identify worst-case disruptions and develop optimized strategies to mitigate their

*Corresponding author. Email: hkavak@gmu.edu. Address: Department of Computational and Data Sciences, George Mason University, 4400 University Drive, MS 6A12, Fairfax, VA 22030-4444

impact. The modeling approach proposed in this paper provides a basis for planning tools that support resilience and preparedness of supply chains.

Keywords: Hybrid Simulation, Supply Chains Modeling, Resilience, Optimization, Evolutionary Computation

1 Introduction

The term supply chain refers to "an integrated manufacturing process wherein raw materials are converted into final products, then delivered to customers" [8]. Due to the globalization of manufacturing operations, most products today are produced through complex supply chain operations distributed worldwide. Judicious sourcing strategies render product development more efficient and cost-effective.

The recent COVID-19 pandemic has brought supply chains and their vulnerability to disruptions front and center. Such disruptions not only led to global shortages of critical goods, such as semiconductors, personal protective equipment, and medical supplies, but also impacted the roll-out of vaccines [16, 47]. Disruption of supply chains due to disasters, such as pandemics, has long been a societal concern [34]. Such disruptions may result not only from natural disasters, but also from other sources. For example, criminal organizations disrupt and manipulate licit supply chains for financial gain, political objectives, etc. [35, 52, 59, 51]. The impact of such criminal activities during a disaster is even more pronounced as criminal organizations take advantage of a chaotic and uncertain environment [44, 25]. Thus there is a strong need for modeling and simulating not only supply chain operations but also malicious actors who may act to disrupt them.

This paper proposes a hybrid simulation methodology, dubbed Hybrid Integrated Supply-Chain Simulation (HISS), to investigate supply chains and their vulnerabilities with the goal of making them strategically and operationally more resilient. The HISS methodology offers a framework for modeling and simulating attacks on real-world supply chains. HISS utilizes the mature modeling platform MASON, an open-source agent-based modeling toolkit with a rich set of visualization and statistical tools, widely used for simulation in many areas [38]. MASON admits both discrete-event simulation (DES) and agent-based simulation (ABS) models in an open-source Java environment [21]. With newly designed DES capabilities, many business processes can be modeled within MASON, while the ABS capabilities facilitate the representation of external factors, including (for example) the dynamics of malicious actors and supply chain maintainers. Researchers can download the packages and develop their own supply chain models to study disruptions.

To showcase the efficacy of the HISS methodology, we modeled and studied the effects of disruptions to a pharmaceutical supply chain due to a disaster (e.g., the COVID-19 pandemic) or through malicious efforts of a criminal organization (e.g., drug adulteration). We chose the pharmaceutical supply chain domain because it has been affected by the COVID-19 pandemic, leading to a shortage of critical pharmaceuticals [12, 47, 16]. Our study further identified robust mitigation strategies to address potential future disruptions.

The contribution of this research to the modeling and simulation community is the HISS methodology and simulation tool, which facilitate supply chain design that is resilient to disruptive events such as pandemics or criminal attacks. It integrates supply chain disruptions and mitigations into the supply chain model, allowing users to study a vast number of potential disruptions before they occur. We also combine ECJ [49] with MASON to simultaneously search for both potent supply chain disruptions and the most effective mitigations thereof. ECJ is a popular open-source stochastic optimization toolkit written in Java with which MASON dovetails well.

Technically, the HISS methodology and simulation tool combine agent-based, DES, and stochastic optimization to model pharmaceutical supply chain production and distribution processes with and without disruptions. Included is an anomaly detection algorithm operating on the simulated data for providing warnings of disruptions, looming or in progress. Our methodology is applicable to modeling a variety of industries or organizations that depend on supply chains. The capabilities of the proposed HISS methodology help provide guidance to supply chain managers in preparing for potential disruptions and in devising corresponding mitigation strategies. Furthermore, the simulation software can be used as a "what-if" tool to assess the impact of disruptions and the efficacy of mitigation strategies, including in table-top exercises. Finally, given that the scenarios presented in this paper are projected ones, the validation of the model was supported by multiple subject domain experts.

The rest of this paper is organized as follows. We first present related work (Section 2), which provides the background for modeling supply chains, disruptions, mitigations, and criminal actors. This is followed by a description of the HISS methodology and a generic pharmaceutical supply chain model, developed, and vetted through guidance from pharmaceutical industry subject matter experts (Section 3). Section 4 describes a set of experiments that explore system-wide impacts of various types of disruptions and the efficacy of various mitigation strategies. Finally, Section 5 discusses conclusions based on our work, along with directions for future work.

2 Related Work

2.1 Modeling and Simulation of Supply Chains

Modeling and simulation have long been integral to the study of supply chains, enabling researchers to analyze complex dynamics and optimize decision-making. Various techniques have been proposed to model supply chains, such as System Dynamics (SD) [23], Graph Networks [9], DES [40, 33, 56], and ABS [60].

The choice of technique is dictated by the purpose of the model and the technique's applicability to the system under study. In situations calling for high-fidelity representation of a system under study as well as associated disruption scenarios and mitigation strategies, hybrid modeling (i.e., using a combination of multiple modeling techniques) is preferred as it can better capture real-life business processes [5]. Simulation modeling offers an effective way of high-fidelity digital modeling and analysis of complex systems, supply chains included. In particular, simulation models provide dynamic computational representations of supply chains, allowing one to study evolving scenarios, such as disruptions, in an in vitro environment.

2.2 Supply Chain Disruption and Mitigation

The complex nature of supply chains makes them vulnerable to disruptions [34]. If not thoroughly understood and managed, supply chain disruptions can lead to significant losses, shortages, panic, and price spikes, among other deleterious outcomes [19]. Since disruptions are common in most supply chains, we aim to develop mitigation strategies to build resilience into them and their operations.

Researchers have studied various strategies of mitigating the negative impacts of disruptions, such as the supply shortages during the COVID-19 pandemic and due to supply-chain attacks by criminal or terrorist actors (e.g., [34, 39, 16, 59]). These strategies include (a) maintaining appropriate levels of safety stocks [2, 27, 28]; (b) improving the visibility of the supply chain; (c) adding real-time sensors at critical points in the supply chain to monitor its status [36]; and (d) diversifying the supplier base [62, 30, 3]. There is also a growing body of research on disruption modeling due to natural and man-made disasters, and their effects on supply chains [54, 14, 42, 32].

2.3 Modeling Criminal Actors Engaged in Supply Chain Disruptions

Most simulation models in the literature focus on supply chain representation, management, and optimization [68, 24, 45]. A handful of simulation research papers model disruptions due to natural causes or accidents [1, 8]. However, there is a growing interest in studying disruptions to supply chains caused by external agents, such as criminal actors [7]. Such disruptions may include material adulteration, physical attacks, theft of ingredients, and cyber-attacks on the supply chain's logistics-support software (e.g., [35, 52, 59, 51]). An earlier defense-focused study [46] lists potential attack points and patterns, and based on an extensive literature review, [48] presents a comprehensive list of measures that can be taken to protect supply chains from intentional attacks. In the same vein, [20] lists micro, meso, and macro drivers of malicious risks associated with supply chains while [29] focuses on the security of digital supply chains.

To date there is limited published simulation research that models the behavior of malicious actors and their impacts on supply chain production and distribution operations. An early example is [65], which focuses on intentional attacks and their simulation using a game-theoretic approach. However, this study has no explicit supply chain or actor representation. In a more realistic setting, [61] investigates various supply chains disruptions including intentional ones, based on simulation of various network (graph) scenarios. Another example is [66], which focuses on automotive supply chain networks with random and intentional attacks. Similar to [61], [66] also provides a simple network model with no actor representation. The extant literature does not consider malicious actors causing disruptions to supply chains and corresponding simulation models. However, the construction of integrated supply-chain simulation models incorporating criminal actors and disruptions, as well as devising mitigation strategies and measuring their efficacy, are complex tasks. This complexity calls for hybrid modeling techniques, addressed by the HISS methodology of this paper. Accordingly, the current paper bridges this gap in the simulation literature by presenting the HISS methodology. This methodology is applicable to a broad range of supply chain disruptions caused by malicious actors.

3 Methodology

Figure 1 depicts a high-level conceptual representation of the HISS methodology, consisting of three main components: (1) MASON's Discrete Event Simulation Extension [22] is used to represent supply chain models; (2) a malicious agent component models disruptive attacks on the supply chain, subject to specified budgetary constraints; (3) a supply-chain manager-agent component devises mitigation strategies to counter the impacts of disruptions. Both the supply chain manager and malicious agent are encapsulated in the ECJ stochastic optimization library [49]. Results from extensive runs of the simulation model are captured for further processing to assess risk and develop early warning systems and mitigation strategies. We next describe the components of the HISS methodology in some detail.

3.1 The MASON Toolkit

MASON [38] is a widely used open-source agent-based modeling toolkit with over 20 years of mature codebase. With its Java-based model implementation, MASON models are designed for speed, flexibility, and ease of integration with other tools. These advantages make MASON an ideal toolkit for complex system modeling and multi-agent simulations, allowing researchers to study complex interactions in dynamic environments. For computationally intensive models, Distributed MASON [18], extends MASON's capabilities by enabling parallel execution across multiple processors in cloud or high-performance computing environments.

One of MASON's distinguishing design features is its separation of the model from its visualization. In MASON, a model is encapsulated within a singleton *SimState* object, which operates independently of visualization or inspection processes. To enable visualization and interactive manipulation, a separate

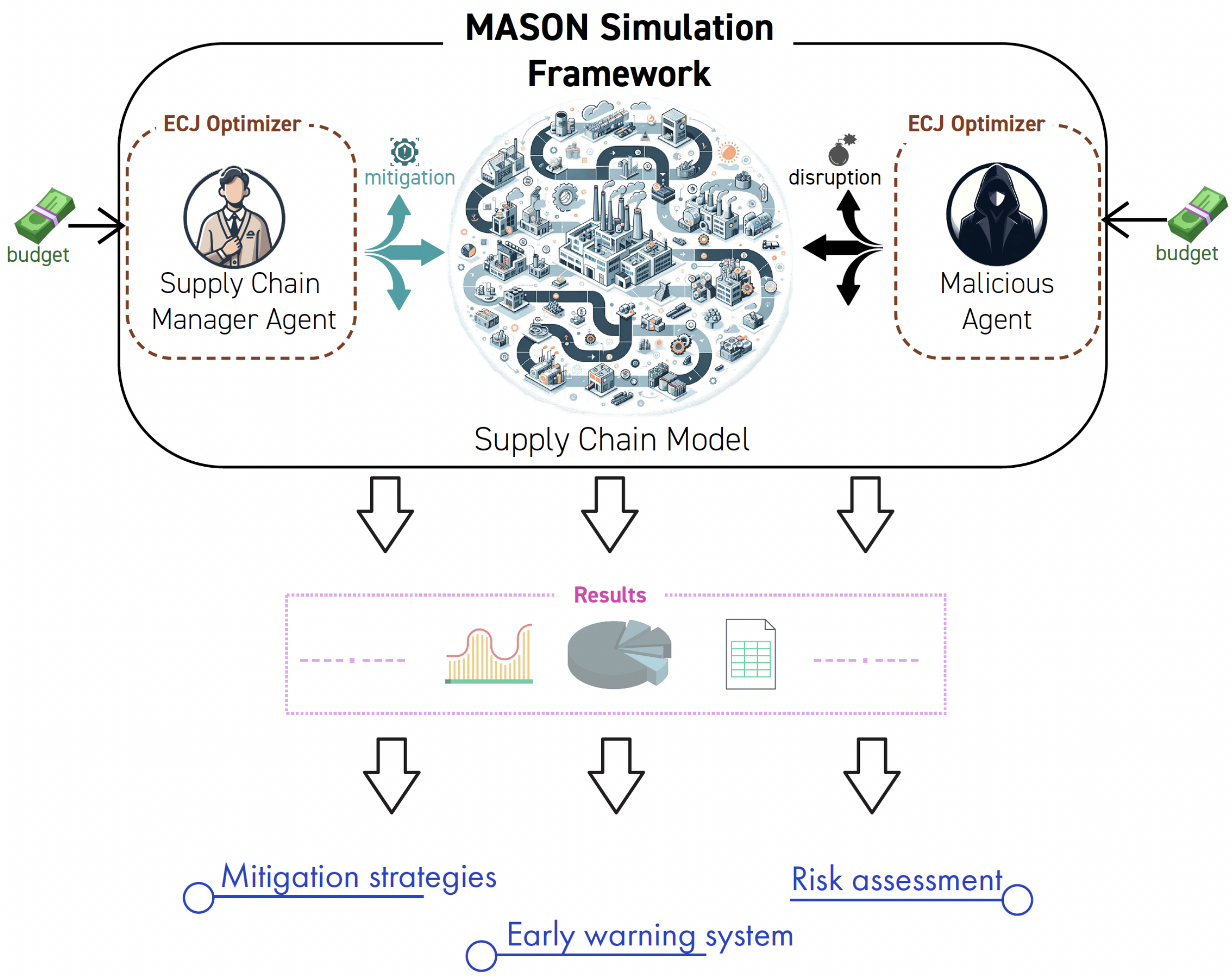

Figure 1: A conceptual representation of the HISS methodology.

GUIState is created, which references *SimState* objects and renders their visual representations. This architecture allows MASON models to function headless (without visualization) faster and to dynamically incorporate different visualization layers as needed. Additionally, it enables models to be checkpointed and transferred to different computing environments to resume the model runs. Due to their self-contained nature, MASON models can execute in parallel within separate threads or within the same thread of a process. This integrates modeling and simulation tasks, such as sensitivity analysis or replications, seamlessly with many instances running at the same time.

MASON models are built upon two fundamental components: *fields* and *time representation*. MASON models utilize *fields* (one or more spatial representations) which define the simulation environment's spatial structure. While the modeler can create custom fields, MASON provides a rich selection of built-in ones, including various grid types, network and graph-based fields, continuous spaces (bounded, unbounded, or toroidal) in two or three dimensions, and GIS-based environments via the GeoMASON extension [55]. *Time representation* in MASON models is implemented using a heap-based schedule which allows individual *agents* to be programmed to activate at predefined future time points to execute specific actions. These agents can dynamically reschedule themselves as needed, enabling flexible and adaptive simulations. Additionally, MASON allows various agent activation regimes to be enabled to govern the order of agents which are scheduled to be executed at the same time.

3.2 The MASON DES Extension

The HISS methodology gives rise to a hybrid model involving both DES and ABS capabilities. DES is a broadly accepted paradigm for modeling of supply chain operations [56, 40]. More specifically, a DES model captures the supply chain's layout as a network, where nodes represent facilities (e.g., production and storage), and edges represent flow pathways of entities, such as raw and intermediate materials, end products, as well as information (e.g., product orders). Entity delays at nodes (e.g., waiting times, production times) and at edges (e.g., transportation times) are typically governed by probabilistic distributions, which the end-user can parameterize. ABS is a modeling paradigm for representing individual actors as agents. It facilitates the dynamic representation of complex interactions involving system actors and other entities across locations and over time.

Complex systems can be modeled using MASON's DES extension [22], which enables hybrid modeling. This extension introduces a graph-based structure where processes interact through resource exchanges in the form of sources, sinks, and filters. Processes function as agents and are scheduled on MASON's agent schedule. This approach optimizes computational efficiency by processing events only when states change. The hybrid system supports macros, which encapsulate complex subgraphs of processes into manageable units. Accordingly, the modeler can integrate macros into higher-level models. This feature is useful for modeling large-scale systems where multiple interconnected components interact dynamically. For instance, in an M/M/1 queue implementation, a simple queuing system is structured using DES components such as sources, locks, delays, and unlocks, where entities flow through the system is driven by stochastic arrival and service time distributions.

This hybrid modeling framework allows researchers to analyze supply chains (among many other systems) and many related operations, including intentional attacks on them. Compared to the widely used commercial hybrid modeling environment AnyLogic [17], MASON offers greater customization despite lacking a visual interface. The extension provides an open-source alternative for researchers developing DES and hybrid ABM-DES models, ensuring flexibility and efficiency in creating and running complex simulations.

3.3 The Supply Chain Model

Modern supply chains are highly complex and that complexity can have an adverse impact on performance [50, 41]. Complexity can be due to the number of parts in production phases, in the number and variety of suppliers across tiers, the variety of customers and the interactions of all actors. In the literature, the number of system components is referred to as structural complexity, and the interaction of these components is referred to as operational complexity [15, 64]. Both complexities, individually and collectively, contribute to uncertainties in the system, making it more difficult to mitigate disruptions. Understanding the system's components and their interactions, is a critical first step in identifying vulnerabilities to disruptions and detecting disruptions in progress.

We next describe a conceptual model of a pharmaceutical supply chain (*PSC*) consisting of sourcing-production-distribution operations. It was constructed in consultation with functional and security experts in the pharmaceutical industry who provided information on the structure of the PSC model as well as data to parameterize it. The developed PSC model was shared with additional industry and government subject matter experts from the pharmaceutical industry. These experts vetted the model, its parameters, and the embedded disruption types, and confirmed its accuracy as a generic representative PSC model for both solid and injectable drugs. All simulation model's parameters are provided in Appendix A1. These include all probability distributions of the model's stochastic components (e.g., production times, transportation times, testing times, etc.). This PSC was then coded as a DES model (dubbed PharmaSim) using MASON. The modeler can modify PSC's topology and parameters, thereby allowing the model's extension to other pharmaceutical supply chains. Source code details are provided in Appendix A1. Figure 2 depicts the layout of the PSC as a supply chain map.

The PSC model consists of a focal company (FC), which is the pharmaceutical manufacturer of interest, and its aggregated external components. Here, all icons marked in gray represent the facilities (nodes) of the FC. The other icons represent entities external to the FC, such as external suppliers, contract manufacturing organizations (CMOs), distributors, wholesalers, and end-customers. The black arrows represent information flows between facilities, while the blue and red arrows represent trusted and untrusted material flows, respectively. Potential disruptions in this supply chain are represented using supply chain exploits prepended with the X letter. The PSC model captures the drug production and testing sequence from raw material to active pharmaceutical ingredient (API), and then to drug production from API and excipients (inert pharmaceutical ingredients), and finally drug packaging. A detailed description of PSC operations is provided in Appendix A2.

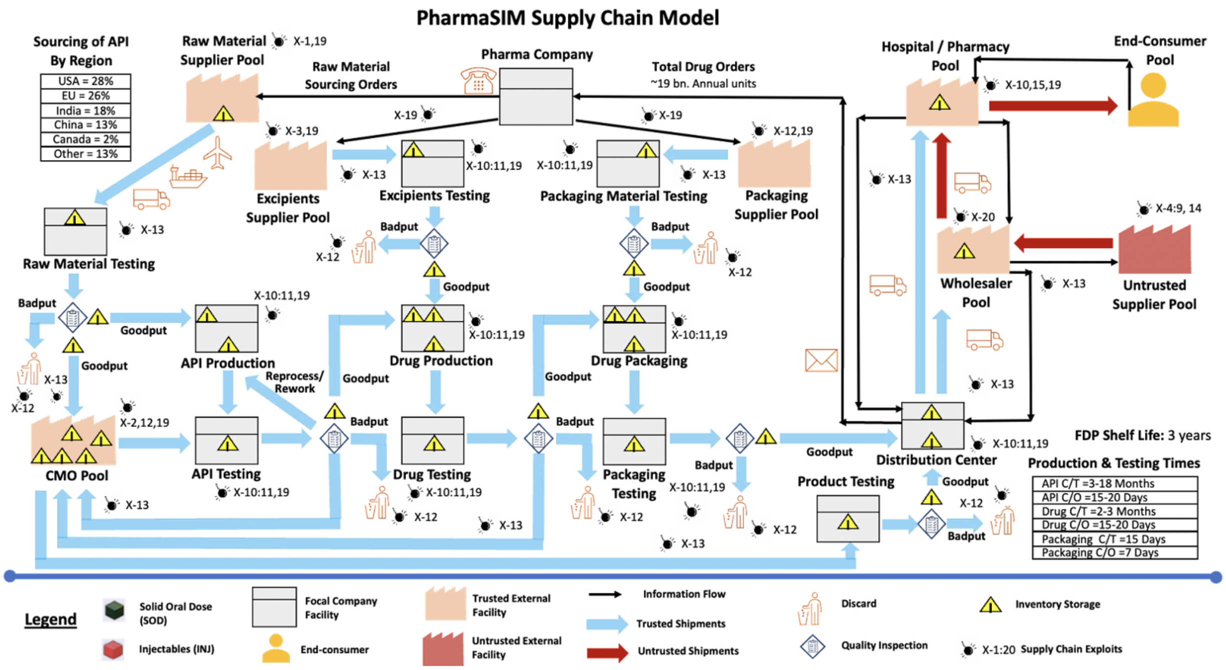

Figure 2: Visual representation of PSC, which was used to code PharmaSim.

3.4 Representation of Disruptions and Mitigations

HISS utilizes optimization techniques to represent criminal actors' attacks on the supply chain to identify worst-case disruptions as well as mitigations from a supply chain agent. HISS identifies a worst-case disruption by computing an objective function that measures the negative impact of a disruption on a performance metric. Every disruption is allocated a fixed budget that is commensurate with the disruption's severity. The budget captures the fact that criminal agents are constrained by finite resources. The criminal agent's optimization problem is to find a maximal-impact disruption (worst-case disruption), subject to the given budgetary constraint. HISS counters such attacks by encapsulating a supply chain agent, which develops a set of mitigation strategies to effectively counteract given disruptions. These strategies involve online anomaly detection, pre-disruption mitigation, and post-disruption mitigation, to be described in the sequel.

When implementing PharmaSim, we made some simplifications. Technically, MASON's *Steppable* interface can be extended to create instances of criminal and supply chain agents that carry out respective tasks. However, our model's disruption and mitigation logic is not complex. They are initially prescribed a new set of parameters, generated after evaluating candidate runs using the optimizer. Therefore, we encapsulated agent representation within the optimizer which sets disruption and mitigation parameters. While the results would be identical to an explicit agent representation, this approach makes experimentation more manageable.

Class	Disruption Code	Disruption Description
1	A2	Non-Receipt of Ordered Raw Materials
2	A3	Raw Material Adulteration
3	A4	Raw Material Destruction
4	A5	API Production Destruction
5	A6	Halt in API Production
6	A7	API Adulteration
7	A8	Non-Receipt of Ordered Excipients
8	A9	Excipient Adulteration
9	A10	Excipient Destruction
10	A11	Drug Production Destruction
11	A12	Halt in Drug Production
12	A13	Drug Adulteration
13	A14	Non-Receipt of Ordered Packaging Materials
14	A15	Packaging Material Adulteration
15	A16	Packaging Material Destruction
16	A17	Packaged Drug Destruction
17	A18	Drug Packaging Halting
18	A19	Packaging Adulteration

Table 1: List of disruptions used in the study (see Appendix A2 for the mapping of disruption and exploit codes).

3.4.1 Types of Disruptions

We modeled various types of commonly occurring disruptions to pharmaceutical supply chains (e.g., [35, 59, 52, 51]). A disruption specification includes a disruption class (e.g., disrupted production, transportation, testing, etc.), a supply chain location (node), and an impacted pharmaceutical product (see Table 1 for the list of disruptions used in the study).

It should be noted that a disruption can result from either non-criminal or criminal causes. For example, non-receipt of ordered supplies could be due to disrupted production or transportation, or due to theft of shipments by criminal agents. Another example is the delayed receipt of shipments which may be due to a decrease in normal production capacity of facilities damaged by a natural disaster or a physical/cyber-attack by a criminal agent. It is important to note that criminal actors can take advantage of some disruptions resulting from natural disasters. For instance, supply shortages due to a disruption may lead distributors to order from untrusted suppliers, resulting in illicit products entering the supply chain. These different causes of disruptions underscore the complex nature of supply chains and show the need for studying a variety of disruptions, including those precipitated by malicious actors as well as those due to other causes.

3.4.2 Online Anomaly Detection (OAD)

Supply chain management faces the challenge of identifying disruptions accurately and expeditiously to take appropriate mitigation actions [63, 57]. To this end, an online anomaly detection (OAD) algorithm was incorporated into the simulation model, which monitors metrics of interest (e.g., daily flow of a product through a point of the supply chain network). OAD reports an in-progress anomaly event on days when such a metric significantly deviates from normal (baseline level). The OAD algorithm is as follows:

1. At the beginning of each day, the average normal level of flow is computed as the average flow over the last 15 days that have not been labeled as "anomalous".
2. At the end of each day, the day's total flow is compared to the current average normal level metric, and if it is less than 80% of that average, the day is labeled as anomalous. The duration and percentage parameters can be changed to match the end-user's requirements.

Since anomaly detection is carried out using daily flows, there is a built-in detection delay of up to 24 hours, depending on the anomaly's start time. In our experiments, we have incorporated disruption-specific OAD units to monitor selected flows in the supply chain. For example, such OAD units measure daily inflows to the inventories of all production nodes of the focal company (e.g., raw material flowing to the API Production node, API and excipients flowing to the Drug Production node, bulk drugs flowing to the Drug Packaging node), as well as the inflows of packaged drugs to the Distribution Center node. A real-world implementation of OAD would require a supply chain management system with various sensors, a central database, and monitoring capabilities.

3.4.3 Post-Disruption Mitigation

Post-disruption mitigation refers to actions taken in response to detected disruptions. Consistent with industry practice and guidance from pharmaceutical experts, the primary post-disruption mitigation strategy in the PSC model was the use of safety stocks [2, 27, 28]. In PharmaSim, an inventory consists of regular inventory (designed to cover average demand) and safety stocks (aiming to protect against stockouts from demand variability, including those due to disruptions). For example, in Figure 2, the API Production node has a safety stocks inventory of raw material, while the Drug Production node has two safety stocks inventories: one of API and the other of excipients. The replenishment policy of all PharmaSim inventories is Make-To-Stock in order to maintain sufficient safety stocks during disruptions and due to the inflexible nature of the manufacturing stage in PSCs [26, 53]. This policy is specified by three parameters: the initial inventory level (which is also the target level), the reorder point, and the replenishment lead time (which may be random from a user-specified distribution). At the beginning of the simulation, all inventory levels are initialized; when an inventory level hits or drops below the reorder point, an order is issued to replenish the safety stock to its target level, and the order arrives after a replenishment lead time. OAD utilizes safety stocks as a post-disruption mitigation as follows:

1. When OAD detects an anomaly in the input flow of a particular ingredient at a production node, the node is notified to use the appropriate safety stocks.

2. When OAD detects that the anomaly has ended, the production node is notified to switch back to using the appropriate regular inventory.

3.4.4 Pre-Disruption Mitigation

Pre-disruption mitigation refers to preparedness actions taken in anticipation of future disruptions. Such preparedness actions are designed to increase the efficacy of expeditious disruption detection and response, and to ensure that points of vulnerability in the supply chain are provided adequate protection. The key pre-disruption mitigations are:

1. Right-sizing of regular and safety stocks inventories.

2. Drawing down on all inventory stocks in their order of arrival to keep them fresh and prevent their expiration.

3. Improving the visibility of the supply chain.

4. Adding real-time sensors at key points that accurately track and report data on the operational status of the supply chain [36, 4].

5. Diversifying the supplier base to facilitate access to alternate backup suppliers, such as CMOs, option contracts, etc. [62, 30, 3].

In particular, improving supply chain visibility facilitates the identification of vulnerability points. Such vulnerability points can be hardened (made more resistant to disruptions) as part of a pre-mitigation effort. The effect of such hardening on criminal agents is to increase the costs of launching disruptive actions.

3.4.5 The Resilience Triangle

In order to quantify a disruption's severity, we use a notion of resiliency, which has continually grown in importance in the field of Supply Chain Management [10, 11, 6]. Accordingly, we leverage the resilience triangle concept [67], as a measure of resilience in the context of disasters. For a given disruption, the resilience triangle measures the resulting performance degradation of a system component (a "social unit" in [13]). Specifically, the height of the resilience triangle represents the severity of the disruption, its base measures the subsequent recovery time, and its area represents the disruptive impact on the system component. In our work, the "social unit" concepts maps to a node representing a supply chain facility of the focal company. The smaller the area of the triangle, the higher the resilience of that node.

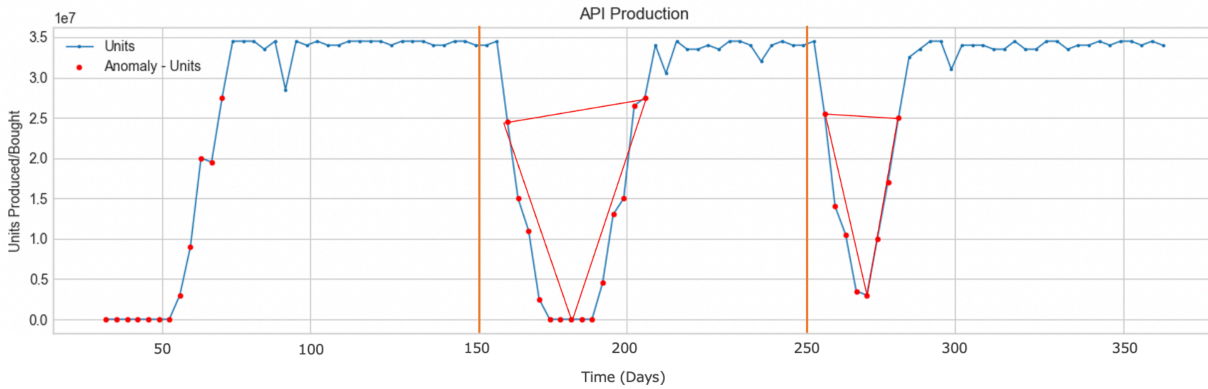

Figure 3: Time series of daily production flow through the API Production node (resilience triangles are shown in red and the number of units on the vertical axis is in millions).

The OAD algorithm constructs the resilience triangle by estimating the start and end time of the disruption in the time series of a metric of interest. For example, Figure 3 depicts a time series of the daily production flow through the API Production node, where two disruptions occur sequentially, the timings of which are indicated by the vertical orange lines. In Figure 3, each of the two resilience triangles is graphically shown as an inverted triangle, where the base is on top and the apex at the bottom. Here, bullets represent discrete measurements of a metric of interest, where the blue dots on the curve correspond to normal behavior (no disruption in progress) and red dots correspond to anomalous behavior detected by the OAD (possible disruptions in progress). Note that the initial sequence of anomalous points (from time 0 until approximately time 70) represents the warm-up period of the simulation, before it reached steady state, and those were ignored in our experiments.

3.5 Optimization with the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)

To identify worst-case disruptions, OAD uses stochastic optimization techniques (sometimes called *meta-heuristics* [37]). Stochastic optimization is a general family of techniques which apply random perturbations to search for solutions in a decision space, and these are very common in scenarios where little is known about the nature of the space itself, and particularly where there is no way to obtain the gradient or higher-order derivatives of the objective function to be optimized.

One of the most common forms of stochastic optimization is the subfamily known as *evolutionary computation*. These are probabilistic resampling methods, inspired loosely by natural selection, of which perhaps the most famous historic example is the Genetic Algorithm [37]. This family is popular because it is readily parallelizable to massively distributed computer systems, and because it is highly general and so can be applied to large and complex problems.

An evolutionary computational algorithm starts with a sample, referred to as a *population*, of randomly-generated or seeded candidate solutions, referred to as *individuals*. The quality (or *fitness*) of each candidate solution is assessed (in our case, the values of performance metrics). Then the population is resampled randomly so as to favor new and fitter solutions.

Among various evolutionary computation methods, the *Covariance Matrix Adaptation Evolutionary Strategy* (or CMA-ES) was chosen by virtue of its status as a well-established and high-performance technique [31]. CMA-ES searches for an optimal solution of a multidimensional, real-valued metric, over an unconstrained decision space. CMA-ES is probably the best-known example of an *estimation of distribution algorithm* (or EDA). An EDA resamples its population by fitting a distribution model over the population, weighted by fitness, and followed by random resampling under that model.

Evolutionary algorithms can be used in a variety of modes. For our purposes, we use an *evolutionary* mode and a *two-population competitive coevolutionary* mode. The evolutionary mode is the algorithm as has been described: a sample (or population) of candidate solutions is iteratively assessed and resampled. In two-population competitive coevolution there are two samples (two populations), each being iteratively assessed and independently resampled. However, individuals in each population are assessed by pitting them against individuals in the other population. For example, in order to develop an optimal sorting network, we might pit a population of candidate sorting networks against a population of sorting problems. The fitness of a candidate sorting network would be based on how well it sorts problems in the other population; but the fitness of a sorting problem would be based on how poorly the sorting networks performed. Thus, as one population is developing better sorting networks, the other population is developing harder sorting network problem to challenge them.

3.5.1 Application

Our evolutionary optimization approach starts with a set of randomly-generated disruptions, and measures their impacts, following which the optimization algorithm produces a new generation of higher-impact disruptions. This process is repeated until a stopping rule is encountered. To this end, the following evolutionary algorithm was used, where N_k is the candidate population at iteration k:

1. Create an initial set of N_0 of randomly-generated candidate disruptions.

2. Repeat k times:

    (a) Test each candidate disruption in N_k and assign it a fitness value.

(b) Apply CMA-ES to resample the set of candidate disruptions in N_k, weighted by fitness, to produce a new sample N_{k+1}.

A disruption candidate is assessed by injecting it into supply chain simulation runs and measuring the arithmetic mean disruption value according to the performance metric of choice.

In the *coevolutionary* approach, we create a population of candidate disruptions and a second population of candidate *mitigations*.

1. Create an initial set of N_0 of randomly-generated candidate disruptions.
2. Create an initial set of M_0 of randomly-generated candidate mitigations.
3. Repeat k times:
   (a) Test each candidate disruption in N_k against sample mitigations in M_k and assign it a fitness value.
   (b) Test each candidate disruption in M_k against sample disruptions in N_k and assign it a fitness value.
   (c) Apply CMA-ES to resample the set of candidate disruptions in N_k, weighted by fitness, to produce a new sample N_{k+1}.
   (d) Apply CMA-ES to resample the set of candidate mitigations in M_k, weighted by fitness, to produce a new sample M_{k+1}.

The fitness of a disruption was based on the *best performance* it had against any mitigation in its test. This was to encourage disruptions to look for outliers that no mitigations had yet covered. On the other hand, the fitness of a mitigation was based on the *mean performance* over all disruptions in its test, to encourage mitigations to become general-purpose over many kinds of disruptions.

3.5.2 Integration with ECJ

HISS was implemented in the ECJ optimization software [49], a popular evolutionary computation and stochastic optimization library. ECJ, like MASON, is over 20 years old, is very widely used, and is open-source and written in pure Java. MASON dovetails well with ECJ: indeed that was one of MASON's original design goals.

ECJ is comprehensive. It contains a large number of evolutionary computation algorithms, including CMA-ES, and many ways to represent candidate solutions and to assess them for quality, including multiple evolutionary and coevolutionary modes. Importantly, ECJ also contains a wide variety of ways to distribute evolutionary computation algorithms over large numbers of machines. We use ECJ's CMA-ES algorithm, evolution and two-population competitive coevolution modes, and its massively distributed evaluation facility, and combine them with MASON, which performs the evaluation portion.

To assess the severity of each disruption and the efficacy of attendant mitigations, the following procedure was used. First, a *master ECJ process* would create candidate disruption scenarios and (if used in coevolutionary mode) candidate mitigations as well. Then the master would iteratively farm out disruptions and (if in coevolution) mitigations to distributed MASON models to assess their performance. After receiving the results of each assessment, ECJ would apply CMA-ES to resample new populations of disruptions and (if in coevolution) mitigations, and then repeat the iteration.

Figure 4 provides an overview of the software implementation of HISS, its components, and their interactions. Specifically, Figure 4 displays the interactions among the ECJ, MASON, and PharmaSim components of the software. Experimental runs utilized a massively distributed optimization framework to search for worst-case disruptions using ECJ. Parallel supply chain model instances were run on multiple MASON processes. ECJ was further used to optimize mitigation strategies. The models ran in a custom DES environment in MASON, visualized using MASON tools, and executed as follows:

1. Hard-coded parameters are set up.
2. The optimizer chooses certain disruption and mitigation parameters and sets/changes them in the model.
3. The model is run and the disruption and mitigation parameters drive "agent" objects to modify the model at runtime as necessary.
4. After the model run is completed, the results are analyzed and statistics are produced.
5. The results go to the optimizer to update it for the next model run.

3.5.3 The Fitness Function

The optimization of worst-case disruptions makes use of an objective function that measures disruption impacts on the supply chain. In this research, the objective function is a *fitness function* that uses the total amount of finished product delivered to the distribution center over the simulation period. This type of fitness function is appropriate since disruptions decrease the delivery of finished products. To ensure that disruption experiments are carried out in supply-chain steady state, the warm-up period was set large enough, such that the standard deviation of the delivered finished product does not exceed a pre-defined threshold. The threshold was determined via a discretionary user-defined parameter that reflects the user's view of an acceptable "degree of stationarity" of the simulation run.

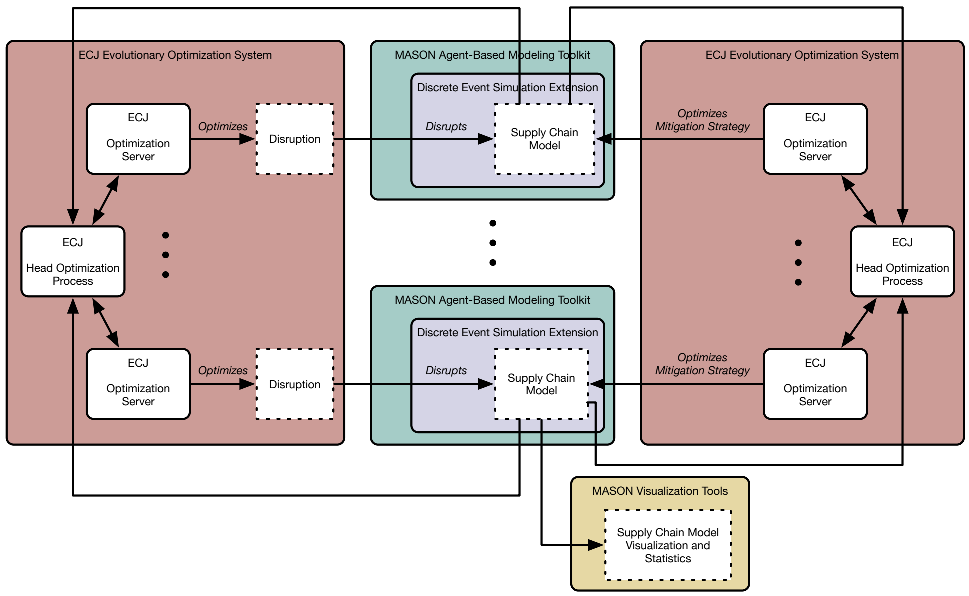

Figure 4: Overview of the software components and their interactions.

4 Experiments and Results

We used PharmaSim as the simulation engine of the PSC model, and applied optimization in multiple experiment sets.

Recall that the optimization's goal sought to attain two objectives: (1) Identifying worst-case disruptions, and their associated vulnerability points; and (2) devising best mitigation strategies. While the results to be reported were obtained from model simulation runs, and as such are hypothetical, the implications of the results are relevant to actual supply chain operations in the field. In what follows we report on injecting single disruptions and multiple disruptions, along with their associated mitigations, into the PharmaSim simulation models, and demonstrate the efficacy of the OAD.

4.1 Single Disruptions

The set of experiments in this section consisted of individual simulations of the 18 disruption classes of Table 1. Each disruption simulation consisted of replications whose length was 365 days. Every disruption was injected 10 days after the warm-up period.

Figure 5 illustrates a disruption scenario, where the transportation of raw material from the Raw Material Supplier Pool node was halted (see upper left graph in Figure 5) causing cascading downstream impacts of this disruption (see remaining graphs in Figure 5). Each graph depicts the total number of units of materials flowing through each node over 365 days. The vertical dashed red lines mark the onset of disruptions. The highlighted red bullets indicate the points in the corresponding time series, identified by the OAD as

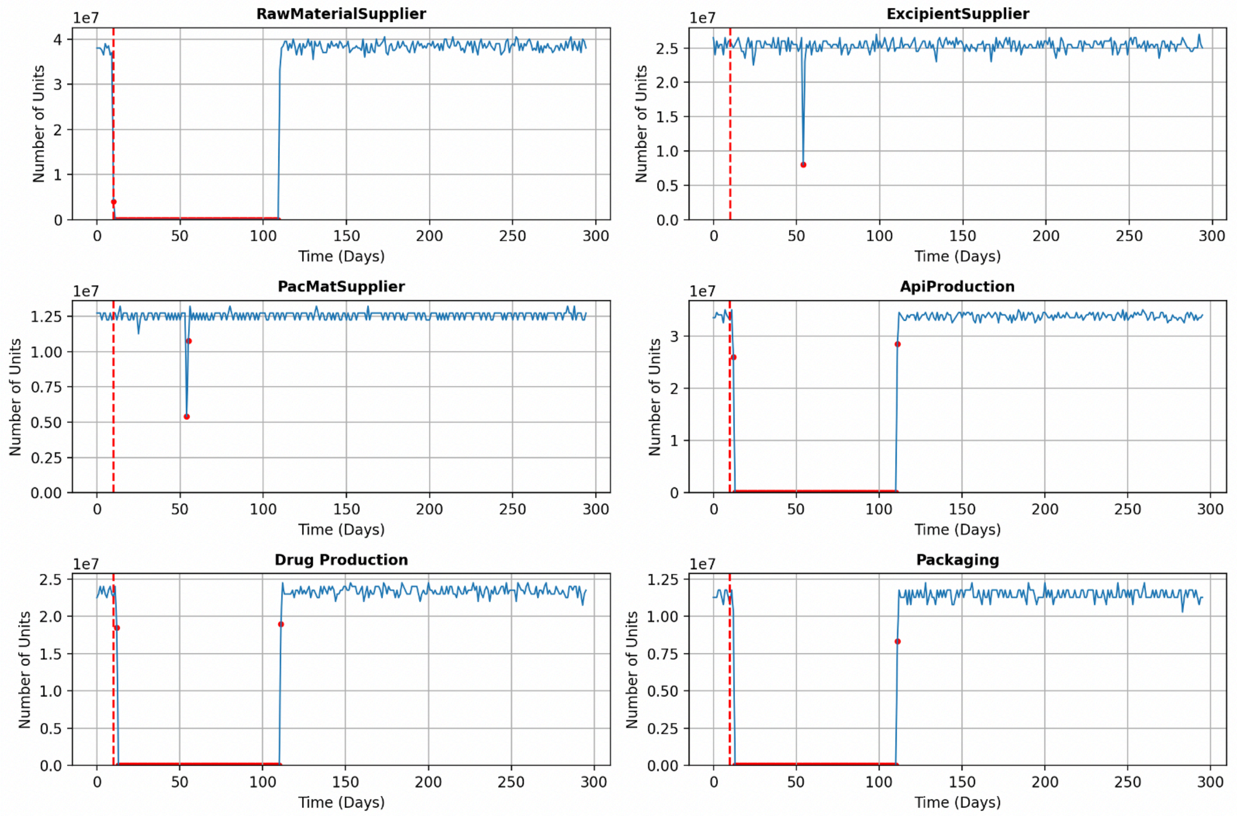

Figure 5: Time series of flows at PSC nodes resulting from halting raw material transportation from the Raw Material Supplier Pool node (the number of units on the vertical axis is in millions).

anomalous.

This particular disruption at the Raw Material Supplier Pool node manifests as a drop in the baseline daily flow through this node. Subsequently, cascading drops in downstream flows manifest at the API Production, Drug Production, and Drug Packaging nodes, with increasing delays. Prompt detection of such cascading disruptions calls for deploying reliable sensors throughout the network. In this instance, if sensors were only deployed at the Drug Packaging node, there would be a 3-day delay in detecting the disruption. This 3-day delay would represent a loss of around 3.45×10^7 units of packaged drugs.

We next analyze the negative impacts of each of the 18 individual disruption classes listed in Table 1. Recall that such impacts were measured by the magnitude (area) of the resilience triangle (see Section 3.4.5) at the Distribution Center node. This node was selected due to its position as the end point of finished products (packaged drugs) produced by both the focal company and the pool of CMOs.

Figure 6 quantifies the impact of various disruptions, using the resilience triangle area metric, and illustrates the variability of impacts across disruption classes. Specifically, non-receipt of ordered materials (disruption classes A2, A8, and A14), and damage to production facilities (disruption classes A6, A12, and

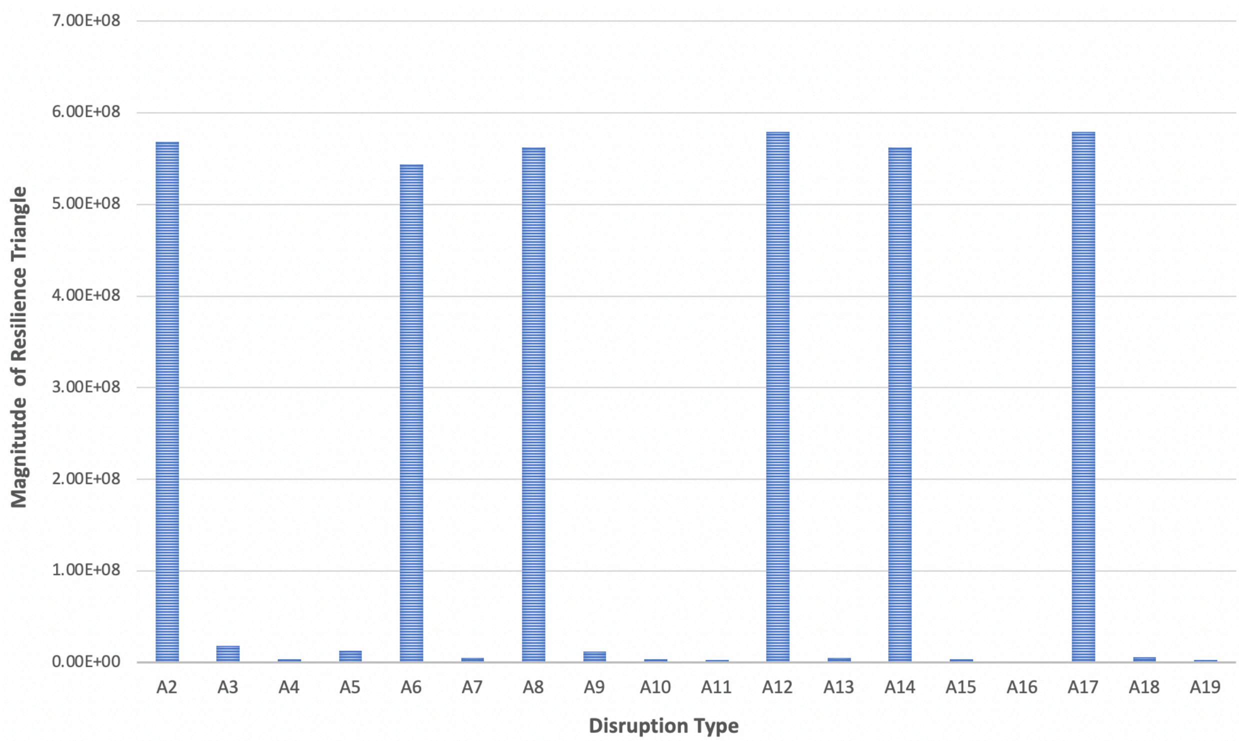

Figure 6: Magnitude of resilience triangles at the Distribution Center node for each of the 18 disruption classes of Table 1 (the vertical axis is in unit-days).

A17), are seen to inflict the most damage. In contrast, adulteration of materials (disruption classes A3, A7, A9, and A15) has a lesser impact as measured by the magnitude of the corresponding resilience triangles. Physical inventory damage (disruption classes A4, A5, A10, A11, A16, and A18) also has a much smaller impact on the supply chain's operations. By their very nature, the first two types of disruptions (non-receipt of ordered materials and damage to production facilities) have a broad effect on the downstream components of the supply chain. However, the next two types of disruptions (adulteration of materials and physical inventory damage) have a largely local effect, as confirmed by the subject matter experts.

4.2 Mitigation of Single Disruptions

The set of experiments in this section analyzed the efficacy of mitigations that increase safety stocks to ameliorate the impacts of supply chain disruptions in Section 4.1. In these experiments, OAD was used to identify where and when a disruption originated, and to trigger tapping of safety stocks as mitigation at the corresponding node in the supply chain network.

Figure 7 displays the experiments' typical levels of safety stocks, which were determined by input from pharmaceutical industry domain experts. The process of determining safety-stock levels for production materials is based on factors such as lead times, material costs, and the number of suppliers. However, the

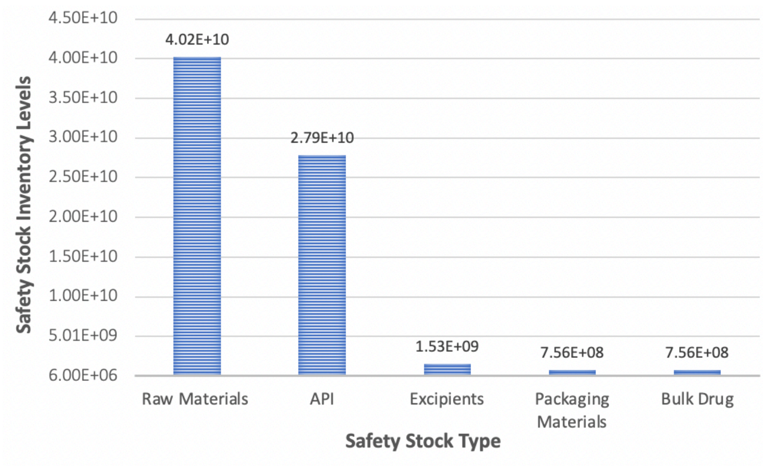

Figure 7: Typical safety-stock levels at key nodes in the PSC model.

experiments to be described later included multiple safety-stock levels, deployment times, and other types of mitigations, discussed in Section 3.4.4. Delays in the detection and deployment of mitigations may result from inaccurate information about the state of the supply chain due to missing or inaccurate sensor data. Thus, the importance of accurate sensors and prompt deployment of safety stocks is evident.

Figure 8 displays the magnitude (area) of the resilience triangle of selected disruptions at the Distribution Center node for three randomly selected disruptions identified in the experiments discussed in Section 4.1. Specifically, it displays the magnitude of resilience triangles resulting from 12 experiments with immediate anomaly detection and 3 classes of disruptions, each subject to 4 scenarios: no safety-stock mitigation (production is resumed after recovery from the disruption), 10-day delay to deploy safety stocks, 5-day delay to deploy safety stocks, and no delay to deploy safety stocks. The results support the efficacy of safety stocks in mitigating the impacts of disruptions to the supply chain. When safety stocks are set at adequate levels, and deployed immediately after an anomaly is detected, the impact of the three randomly selected disruptions (A2, A6, and A12) are adequately mitigated. Even with a 10-day detection delay, the impact of these disruptions is reduced by a factor of about 5. This again illustrates the importance of having reliable sensor data at all points in the supply chain and sufficient safety-stock levels to mitigate disruptions.

4.3 Multiple CMA-ES Optimized Disruptions

This section describes experiments that optimize disruptions to the supply chain, as outlined in Section 3.4, but unlike the single-disruption case, the experiments here consider multiple concurrent disruptions to the supply chain. The goal is to compare the cascading impacts of such multiple disruptions to single-disruption counterparts, subject to increased budgets available to malicious agents. The choice of disruption types and their times of occurence are decision variables. The CMA-ES optimizer identifies not only the worst-case

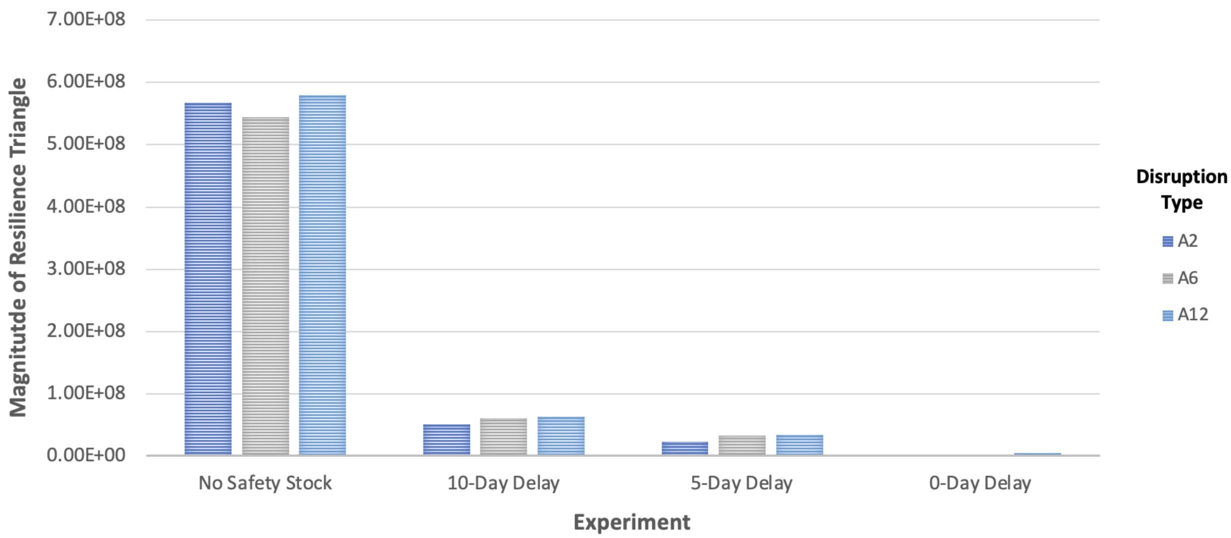

Figure 8: Magnitudes of resilience triangles of three randomly selected disruptions at the Distribution Center node (the vertical axis is in unit-days).

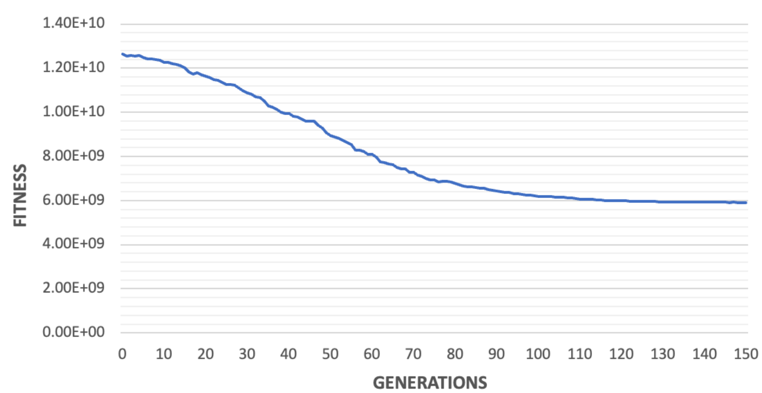

Figure 9: Optimal fitness values returned by the CMA-ES optimizer subject to a fixed budget available to malicious agents.

disruptions, but also their timings. Again, the severity of disruption impacts is measured by the fitness function of Section 3.5.3.

Figure 9 illustrates the improvement in the fitness values, returned by the optimizer. The CMA-ES optimizer was run for 150 generations under a fixed budget available to the malicious agents. The figure illustrates the efficacy of the worst-case disruption optimization, manifested as a progressive decrease of the supply chain's fitness value. CMA-ES converged to its best resource allocation around generation 110, with only negligible improvement thereafter.

Table 2 displays the optimal resource allocation of a budget of 200 resource units to the 18 disruption classes. CMA-ES's best disruption timing shows that the bulk of the budget's resources (161 out of the

Disruption Type	*Resources Allocated*
A2	161
A3	0
A4	5
A5	3
A6	1
A7	11
A8	2
A9	1
A10	1
A11	2
A12	0
A13	2
A14	2
A15	2
A16	3
A17	1
A18	1
A19	2

Table 2: Optimized disruption resource allocation of a fixed budget of 200 resource units.

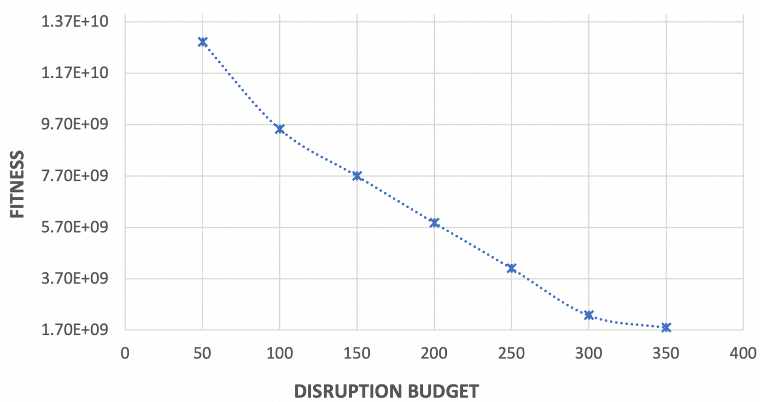

Figure 10: Supply-chain fitness values as function of the disruption budget.

available 200 resource units) was allocated to disruption class A2, which disrupted the transportation of raw materials from the Raw Material Supplier Pool node to the Raw Material Testing node of the focal company in Figure 2. Indeed, note that this allocation gives rise to a major disruption which blocks the operation of both the focal company's and CMO's production process flows. On the other hand, a disruption elsewhere has a comparatively minor impact on supply chain flows.

Figure 10 illustrates the supply chain's fitness values for increasing disruption budgets available to malicious agents. It shows how increases in the disruption budget decrease the fitness of the supply chain in the form of supply-chain's performance degradation.

Finally, Figure 11 illustrates the optimal budget allocation to disruptions at increasing budget levels. The optimizer identified the Raw Material Supplier Pool node as the attack point yielding the most degradation of

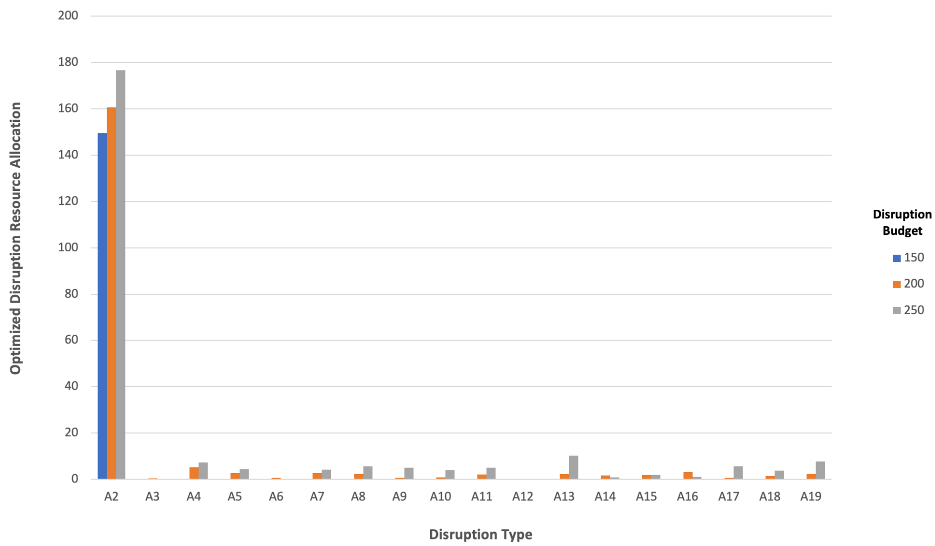

Figure 11: Optimized budget allocations to disruptions at increasing budget levels.

supply-chain performance across all budget levels. Importantly, identifying the most vulnerable points of the supply chain facilitates devising corresponding pre-disruption mitigation strategies (see Section 3.4.4). While some attack points may comport with intuition, we found that SMEs can have differing opinions on the worst points of vulnerability. In such cases, the simulation tool can provide data-driven guidance on this matter.

4.4 Mitigation of Multiple CMA-ES Optimized Disruptions via Safety Stocks

The set of experiments in this section investigated how safety stocks sizing can be optimized to mitigate the worst-case disruptions identified in Section 4.3. In this set of experiments, the safety stocks were deployed without delay after an anomaly has been detected by the OAD.

Figure 12 illustrates the fitness values of optimized safety-stock sizes for increasing safety-stock budgets, subject to optimized disruptions at various disruption budget levels. The results show the efficacy of safety stocks in mitigating worst-case disruptions, identified by the CMA-ES optimizer. This figure further underscores the importance of having reliable sensors and adequate safety stocks at all points in the supply chain to allow timely deployment of requisite safety stocks.

Figure 13 illustrates the optimal safety-stock sizing at various points in the supply chain for worst-case disruptions, at various disruption-budget levels, which in turn reflect different optimal disruption schemes for different budget levels. Again, this figure reaffirms the importance of having adequate safety stocks across all points in the supply chain as a mitigation strategy. The experiments in this section demonstrate the usefulness of the CMA-ES optimizer in identifying points of vulnerability in the supply chain as well as corresponding optimized mitigations.

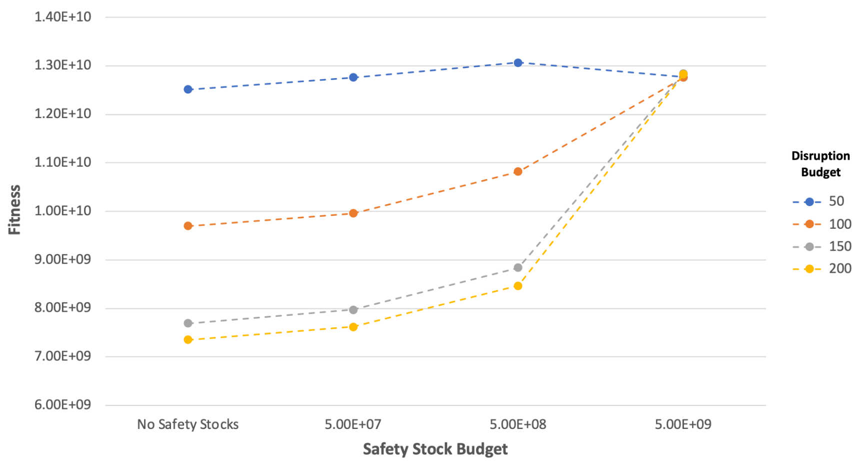

Figure 12: Fitness values as function of increasing safety-stock budgets for various disruption budget levels.

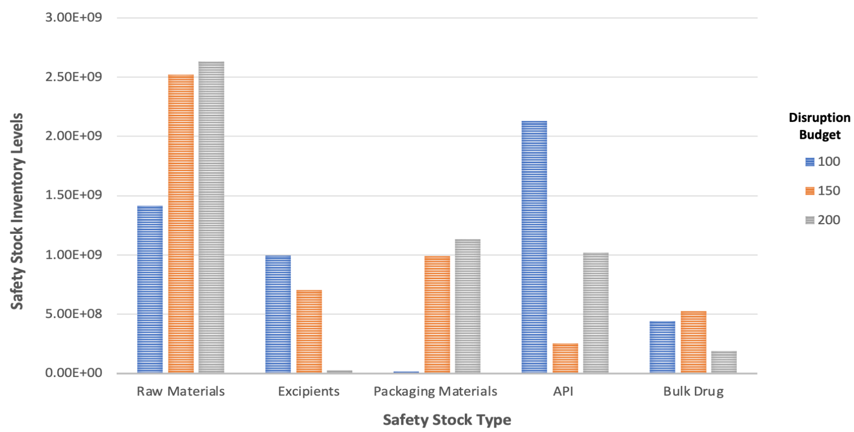

Figure 13: Optimal safety-stock sizing at various supply chain points for various disruption-budget levels.

4.5 Multiple CMA-ES Optimized Disruptions for Given Pre-Disruption Mitigation Strategies

The set of experiments in this section explored CMA-ES optimization of worst-case disruptions, given that pre-mitigation strategies (see Section 3.4.4) have been deployed at the Raw Material Supplier Pool node and API Production node. These nodes were selected because they were identified as points of vulnerability in Section 4.3. In the first experiment, pre-disruption mitigation strategies make these two facilities twice as difficult to disrupt by increasing the cost of injecting disruptions at these points. This notion of "increased difficulty to disrupt a node", can be referred to as hardening the node, which is accomplished by scaling the disruption optimizer's resource budgets by a weight vector prior to invoking the CMA-ES optimizer. As an illustration, Figure 14 compares the new optimized disruption's budget allocation (weighted) to its original

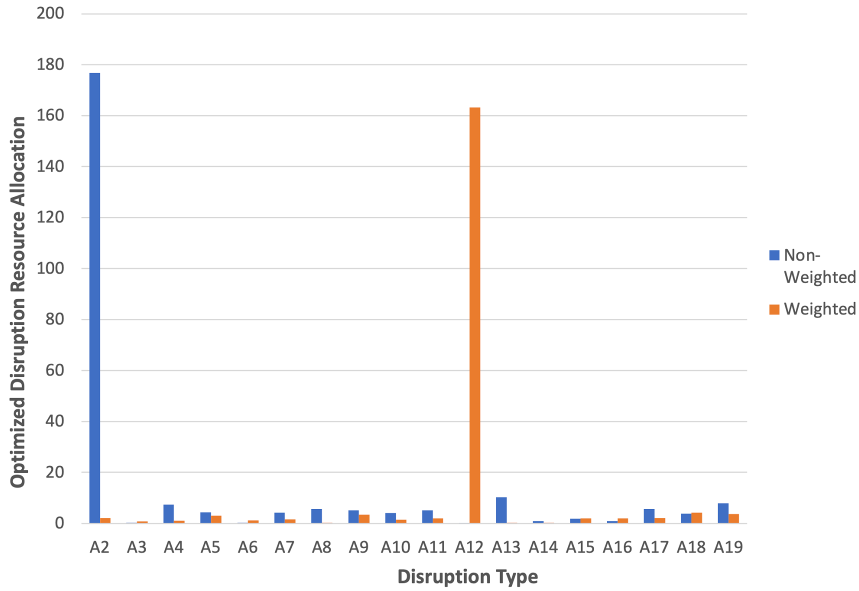

Figure 14: Comparison of weighted and non-weighted optimized disruption resource-budget allocation, when two nodes are hardened, for a fixed budget of 200 resource units.

counterpart (non-weighted), at a total budget of 200 resource units.

Note that once the pre-disruption mitigation strategies were deployed, the Raw Material Supplier Pool node and API Production node are no longer the most vulnerable points in the supply chain. Rather, disruption A12, which targets the Drug Production node in Figure 2, now becomes the most vulnerable point, and its disruption has the most detrimental impact on performance. This insight was not evident before pre-disruption mitigation strategies were deployed, thereby supporting the efficacy of this approach.

Figure 15 illustrates the effect on the optimized disruption budget allocation when the Drug Production node is also hardened in addition to the already hardened Raw Material Supplier Pool and API Production nodes. Here, the most vulnerable point is now the Excipients Supplier Pool node, subject to Disruption A8. In this supply chain, the most vulnerable points in prioritized order of hardening are the nodes Raw Material Supplier Pool, API Production, and Excipients Supplier Pool. Such prioritized hardening guides supply chain managers in allocating limited pre-disruption mitigation resources, subject to budgetary constraints. The examples in this section illustrate the value of optimization methods, such as CMA-ES, in prioritizing vulnerable points to be hardened by pre-mitigation strategies across the supply chain.

4.6 Efficacy of the OAD

This section investigates the efficacy of the OAD, described in Section 3.4.2. Table 3 presents the performance of the OAD in detecting the occurrence of disruptions in the experiments of the previous sections. The definition of these statistics is as follows.

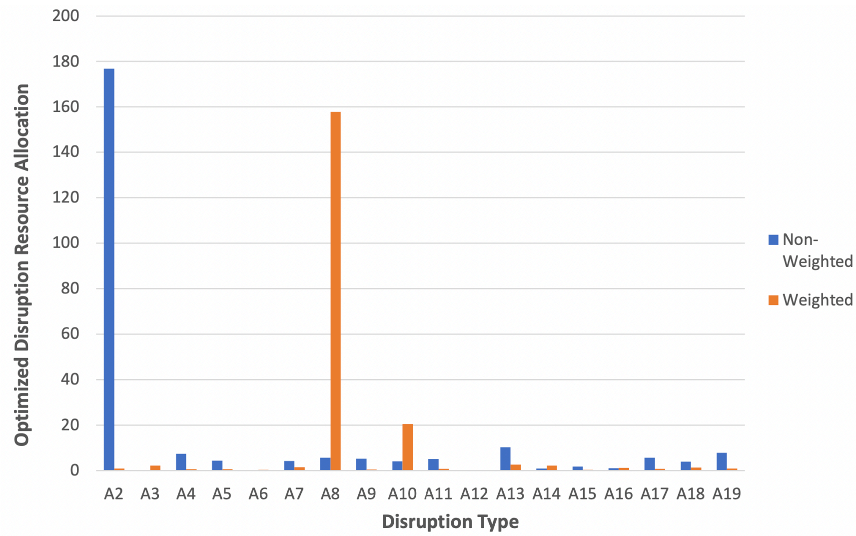

Figure 15: Comparison of weighted and non-weighted optimized disruption resource-budget allocation, when three nodes are hardened, at a fixed budget of 200 resource units.

Statistic	*Score*
Recall	0.966
Precision	0.748
F1 Score	0.825

Table 3: Summary statistics of the detection efficacy of the OAD.

$$Recall = \frac{TP}{TP + FN} \qquad Precision = \frac{TP}{TP + FP} \qquad F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where TP refers to True Positive, FN refers to False Negative, FP refers to False Positive, and $F1$ is the harmonic mean of precision and recall.

The results show a high Recall statistic of 0.966, which implies high performance in detecting disruptions once they have occurred. The Precision statistic is lower at 0.748, suggesting that some of the detected anomalies are not caused by disruptions but are a result of normally occurring random fluctuations. Note that the OAD cannot always distinguish between anomalies due to normal random fluctuations and anomalies caused by certain classes of disruptions, such as adulterated excipients received by the Drug Production node from the Excipients Supplier Pool node.

Figure 16 displays two time series of the number of packaged units over time. The left-hand time series exhibits a single anomaly due to a disruption, while the right-hand time series exhibits two anomalies due to normal fluctuations, both at the Excipients Supplier Pool node. The occurrence time of the disruption in the left-hand time series is marked by a vertical red dashed line, and the same time is marked also in the right-hand time series, where no disruption anomaly occurred. Since the cause of anomalies is generally

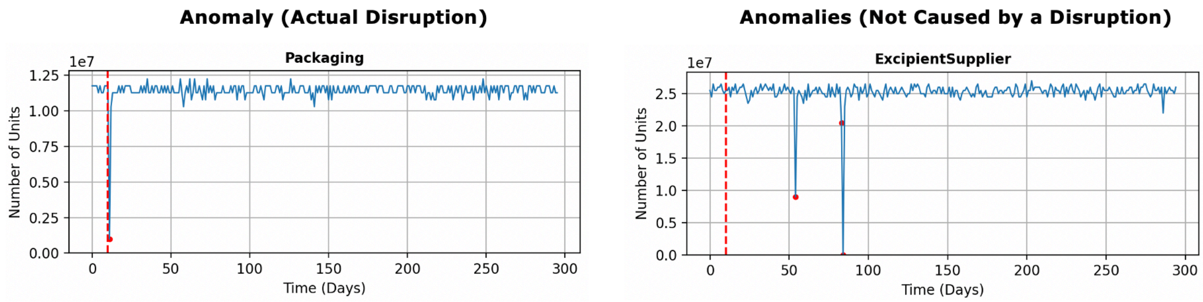

Figure 16: Sample time series of numbers of packaged units with anomalies due to (left) a disruption and due to (right) normal fluctuations (the number of units on the vertical axis is in millions).

Node	*Mean Detection Time Lag (in Days)*
Raw Material Supplier Pool	0.08
API Production	1.04
Drug Production	2.31
Drug Packaging	3.45
Distribution Center	4.17

Table 4: Average time lag of detecting an upstream disruption at downstream nodes by the OAD.

difficult to determine, the Precision statistic produced by OAD tends to be lower than the Recall statistic. Still, all anomalies and their probable causes merit investigation, and the high Precision statistic of the OAD renders it a useful anomaly detection tool. As was shown in the previous sections, OAD deployment, coupled with adequate levels of safety stocks, can mitigate most disruption types, including worst-case ones.

Table 4 reports the average time lag between the occurrence of an upstream disruption (at the Raw Materials Supplier Pool node) and its detection at various downstream nodes in the supply chain. The results indicate the efficacy of OAD in terms of the progression of time lags from the node of disruption to detection at downstream nodes of the supply chain. These statistics further underscore the importance of deploying reliable sensors across the supply chain and the critical need to protect these sensors from damage or cyber-exploits, as such sensors are critical for the timely deployment of mitigations. For example, if a disruption occurred at the Raw Material Supplier Pool node and sensors were only deployed at the Drug Packaging node, then it would take over three days on average before the disruption is detected and mitigated with a loss of around 3.45×10^7 in packaged drug end-units.

5 Conclusion and Future Work

This study presented the HISS methodology — a novel hybrid simulation approach which integrates DES and ABS, coupled with stochastic optimization, applied to supply-chain networks in order to study and mitigate disruptions to them. The efficacy of this approach was showcased through a generic pharmaceutical supply-chain model that integrated disruptions into a supply chain map. This model was developed with input from an advisory group of subject matter experts, and the impact of various disruption and mitigation scenarios was assessed. The pharmaceutical supply chain model consists of common supply-chain elements that could be reused or readily adapted to other supply chain models. Similarly, model parameters showcase a generic supply chain which can easily be modified.

The novel combination of DES and ABS in MASON provides enhanced capabilities that neither modeling approach can provide alone. DES provides representations of supply chain models while ABS provides flexible facilities for modeling disruption/mitigation actors. We encapsulate the disruptive behavior of malicious agents and the mitigation behavior of supply chain agents within a stochastic optimization approach called Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES).

Note that the OAD model used in this paper could be enhanced with a machine learning component if an adequate number of ground truth samples were available for training purposes. Furthermore, the accuracy measures used in the paper pertain to the simulation experiments, where the ground truth was generated by the optimizing module of the simulation. As a direction for future research, the efficacy of the HISS methodology would be further supported by real-life data in addition to simulation experiments.

The MASON toolset is open-source, and researchers can reuse components to create and study their own supply chain models including sets of agents that aim to disrupt or mitigate them. Using the HISS methodology and simulation tool allow users to identify supply-chain vulnerability points susceptible to disruptions, and to assess the efficacy of attendant mitigation strategies. To this end, evolutionary and coevolutionary optimization algorithms were run in parallel. Key findings from the simulation runs indicate that preparedness strategies, such as maintaining adequate safety stocks and enhancing real-time monitoring, significantly improved supply-chain resilience to disruptions. Thus, the HISS methodology advances modeling and simulation and provides significant contributions to build supply chain production and distribution processes that are resilient to disruptive natural events and criminal attacks.

Several future work directions may be pursued. One immediate direction would be to enhance agent representation by individually implementing a set of agents that collaborate with each other in disruption and mitigation actions. One can also explore enhanced anomaly detection capabilities to reduce disruption detection errors. Another direction is to leverage machine learning techniques in predicting disruptions before they occur, thereby strengthening supply chain defenses. Finally, the HISS methodology can be extended to other supply chains and other industries beyond pharmaceuticals.

Disclosure of interest

No potential conflict of interest was reported by the author(s).

Appendices

Appendix A1. Model Parameters and Source Code

- All PSC model parameters are available at https://ccicada.org/2025/06/03/parameters-for-a-generic-pharmaceutical-supply-chain-model/.
- The hybrid implementation of MASON is part of the main MASON package, available at https://github.com/eclab/mason.
- ECJ, including its CMA-ES, coevolution, and massively distributed stochastic optimization facilities, is available at https://github.com/eclab/ecj.
- The use case presented in this paper is available at https://github.com/eclab/DES-Supply-Chain-demo under the name Supply Chain 1 or SC-1.

Appendix A2. Detailed Description of the PSC Model

This appendix supplements Section 3.3 with more details on the pharmaceutical supply chain (PSC) model, depicted in Figure 2. The PSC model parameters may be found at https://ccicada.org/2025/06/03/parameters-for-a-generic-pharmaceutical-supply-chain-model/.

There are two PSC drug categories: pills (Solid Oral Dose, abbreviated as SOD) and injectables (abbreviated as INJ). The Hospital/Pharmacy Pool node receives daily orders of drugs from the End-Consumer Pool node. The sizes of these orders are drawn from triangular distributions following supply chain models in the pharmaceutical sector [58, 43]. The Hospital/Pharmacy Pool node is replenished by the Distribution Center node or the Wholesaler Pool node, and replenishment orders follow a *Make-To-Stock (MTS)* policy (i.e., whenever the inventory drops below its reorder point s, it places a replenishment order that brings the inventory to its target level S). The Wholesaler Pool node is similarly replenished from the Distribution Center node and the Untrusted Supplier Pool node. All replenishment lead times are random variables following pre-specified distributions. If the Distribution Center node lacks sufficient on-hand inventories to fulfill an order from the Hospital/Pharmacy node, it will ship the available on-hand amount, and the rest (if any) is fulfilled by the Wholesaler Pool node. However, if the Wholesaler Pool node lacks sufficient on-hand inventories to fulfill an order from the Hospital/Pharmacy node, it will ship the available on-hand amount, and the rest (if any) is fulfilled by the Distribution Center node or the Untrusted Supplier Pool node, representing suppliers who have not been properly vetted.

The Distribution Center node maintains order Information from the Hospital/Pharmacy Pool and Wholesaler Pool nodes and computes the previous 4-month averages of the aggregate orders for each drug category. The Distribution Center node then sends to the Pharma Company node a monthly replenishment order for each drug category whose size is the corresponding 4-month average. The Pharma Company node converts the received orders into batches of raw materials, excipients (inactive ingredients such as preservatives or coloring agents), and packaging materials, and places corresponding monthly orders with the Raw Material Supplier Pool, Excipients Supplier Pool, and Packaging Supplier Pool nodes, respectively. Upon receiving an order, each of these suppliers produces and transports its ordered supplies in batches to the appropriate node of the focal company. Each batch is then tested at the respective testing facility of the focal company (quality inspection) with a testing delay following a user-specified probability distribution and a defect percentage. Throughput material that passes testing will be referred to as goodput, while failed (defective) material (discarded rejects) will be referred to as badput.

Production can be carried out at the focal company or outsourced to a CMO (contract manufacturing organization) in the aggregate CMO Pool node. Incoming raw material goodput is split between the focal company and the CMO Pool node using user-specified probabilities. Raw material goodput at the focal company proceeds to the API Production node without delay. This node produces API (Active Pharmaceutical Ingredient) in batches from raw material with a user-specified production delay distribution. The API batches are then tested, with three possible outcomes from a user-specified distribution: pass (goodput), fail (badput), or rework (badput). The goodput API is subject to a user-specified split between the focal company's Drug Production node and the CMO Pool node. The former is mixed with the excipient goodput at the Drug Production node to produce drugs. Similarly, the produced drugs proceed to be tested, and drug goodput is also split between the focal company's Drug Packaging node and the CMO Pool node. Drug goodput is matched with packaging material goodput at the Drug Packaging node to yield finished products. All finished products are further tested before shipping to the Distribution Center node. Note that raw material, API, and drugs are processed, tested, and moved as batches. Batch testing is "all or nothing", that is, it yields either a goodput batch or a badput batch (to be discarded). However, when packaging is involved, testing is of individual drug end-units in a batch, and results in the removal (and discarding) from the batch, of the badput end-units only, while the remaining goodput is then allowed to proceed to the next destination. Also, the API Production, Drug Production, and Drug Packaging nodes have safety-stock inventories of their incoming ingredients, which are continually refreshed by using their units in their first-in-first-out order of arrival to the inventory. Every safety-stock inventory uses the *MTS* inventory replenishment policy with user-specified parameters.

The CMO Pool node receives a portion of the raw material goodput from the focal company and splits it (with user-specified percentages) into two groups: raw material to produce API to be shipped back to the API Testing node at the focal company, and raw material for producing packaged drug products. The CMO Pool node further receives from the focal company a portion of the API goodput and a portion of the drugs' goodput and converts them into packaged drug products. All finished products produced by the CMO Pool node are shipped to the Product Testing node and the corresponding goodput is stored in the Distribution Center node at the focal company.

It should be pointed out that unlike models encompassing the full extent of supply chains, this research focuses on understanding the impact of disruptions on a focal pharmaceutical company through detailed modeling of its production, testing, and packaging processes. The demand, supply sourcing, work outsourcing, and distribution components of the supply chain have been aggregated to maintain adequate fidelity while keeping the system easy to parameterize and understand. The innovative feature of the PSC model is the integration of the focal company's facilities and its external components (demand, sourcing, distribution, etc.) with a broad variety of external and internal disruptions. This feature allows the user to readily measure disruption impacts on the focal company's ability to produce drugs and fulfill customer demand.

Appendix A3. Disruption-Exploit Code Mapping

Table 5 lists the PSC disruptions implemented in this research, along with their exploit codes and descriptions.

Scenario	Disruption Code	Disruption Description	Exploit Code	Exploit Description
1	A2	Raw Material Theft	X-13	Transportation Theft
2	A3	Raw Material Adulteration	X-1	Raw Material Adulteration
3	A4	Raw Material Destruction	X-10	Intentional Destruction
4	A5	API Production Destruction	X-10	Intentional Destruction
5	A6	API Production Halting	X-19	Cyber Exploit
6	A7	API Adulteration	X-2	API Materials Adulteration
7	A8	Excipient Theft	X-13	Transportation Theft
8	A9	Excipient Adulteration	X-3	Excipient Adulteration
9	A10	Excipient Destruction	X-10	Intentional Destruction
10	A11	Drug Production Destruction	X-10	Intentional Destruction
11	A12	Drug Production Halting	X-19	Cyber Exploit
12	A13	Drug Adulteration	X-4b	Counterfeit product
13	A14	Packaging Material Theft	X-13b	Transportation Theft
14	A15	Packaging Material Adulteration	X-6b	Counterfeit packaging
15	A16	Packaging Material Destruction	X-10	Intentional Destruction
16	A17	Packaged Drug Destruction	X-10	Intentional Destruction
17	A18	Drug Packaging Halting	X-19	Cyber Exploit
18	A19	Packaging Adulteration	X-7	Counterfeit Product in Relabeled Packaging

Table 5: PSC Disruptions and their exploit codes.

Acknowledgment

This project has been funded in whole or in part with Federal funds from the Department of Homeland Security under BOA No. 70RSAT18G00000001, task order no. 70RSAT21FR0000127. The content of this publication does not necessarily reflect the views or policies of the Department of Homeland Security, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government. The "Supply Chain Model" image inside Figure 1 is generated using ChatGPT 4o's DALL·E tool.

References

[1] R. Aldrighetti, I. Zennaro, S. Finco, and D. Battini. Healthcare supply chain simulation with disruption considerations: A case study from northern italy. *Global Journal of Flexible Systems Management*, 20(Suppl 1):81–102, 2019.

[2] B. Amirjabbari and N. Bhuiyan. Determining supply chain safety stock level and location. *Journal of Industrial Engineering and Management (JIEM)*, 7(1):42–71, 2014.

[3] R. Anupindi and R. Akella. Diversification under supply uncertainty. *Management Science*, 39(8):944–963, 1993.

[4] J. Astill, R. A. Dara, M. Campbell, J. M. Farber, E. D. Fraser, S. Sharif, and R. Y. Yada. Transparency in food supply chains: A review of enabling technology solutions. *Trends in Food Science & Technology*, 91:240–247, 2019.

[5] C. Barbosa, C. Malarranha, A. Azevedo, A. Carvalho, and A. Barbosa-Povoa. A hybrid simulation approach applied in sustainability performance assessment in make-to-order supply chains: The case of a commercial aircraft manufacturer. *Journal of Simulation*, 17(1):32–57, 2023.

[6] A. Barroso, V. Machado, H. Carvalho, and V. C. Machado. Quantifying the supply chain resilience. *Applications of Contemporary Management Approaches in Supply Chains*, 13:38, 2015.

[7] G. Basu. The role of transnational smuggling operations in illicit supply chains. *Journal of Transportation Security*, 6(4):315–328, 2013.

[8] B. Behdani, Z. Lukszo, and R. Srinivasan. Agent-oriented simulation framework for handling disruptions in chemical supply chains. *Computers & Chemical Engineering*, 122:306–325, 2019.

[9] M. A. Bellamy and R. C. Basole. Network analysis of supply chain systems: A systematic review and future research. *Systems Engineering*, 16(2):235–249, 2013.

[10] M. Bevilacqua, F. Ciarapica, and G. Marcucci. Supply chain resilience triangle: The study and development of a framework. *International Journal of Economics and Management Engineering*, 11(8):2046–2053, 2017.

[11] M. Bevilacqua, F. Ciarapica, and G. Marcucci. A modular analysis for the supply chain resilience triangle. *IFAC-PapersOnLine*, 51(11):1528–1535, 2018.

[12] B. Bonadio, Z. Huo, A. A. Levchenko, and N. Pandalai-Nayar. Global supply chains in the pandemic. Technical report, National Bureau of Economic Research, 2020.

[13] M. Bruneau, S. E. Chang, R. T. Eguchi, G. C. Lee, T. D. O'Rourke, A. M. Reinhorn, M. Shinozuka, K. Tierney, W. A. Wallace, and D. Von Winterfeldt. A framework to quantitatively assess and enhance the seismic resilience of communities. *Earthquake Spectra*, 19(4):733–752, 2003.

[14] N. Bugert and R. Lasch. Supply chain disruption models: A critical review. *Logistics Research*, 11(5):1–35, 2018.

[15] C.-Y. Cheng, T.-L. Chen, and Y.-Y. Chen. An analysis of the structural complexity of supply chain networks. *Applied Mathematical Modelling*, 38(9-10):2328–2344, 2014.

[16] P. Chowdhury, S. K. Paul, S. Kaisar, and M. A. Moktadir. COVID-19 pandemic related supply chain studies: A systematic review. *Transportation Research Part E: Logistics and Transportation Review*, 148:102271, 2021.

[17] A. Company. AnyLogic: Simulation Modeling Software Tools & Solutions for Business — anylogic.com. `https://www.anylogic.com/`. [Accessed 12-03-2025].

[18] G. Cordasco, V. Scarano, and C. Spagnuolo. Distributed mason: A scalable distributed multi-agent simulation environment. *Simulation Modelling Practice and Theory*, 89:15–34, 2018.

[19] K. F. Davis, S. Downs, and J. A. Gephart. Towards food supply chain resilience to environmental shocks. *Nature Food*, 2(1):54–65, 2021.

[20] S. DuHadway and S. Carnovale. Malicious supply chain risk: A literature review and future directions. *Revisiting supply chain risk*, pages 221–231, 2019.

[21] G. D'Ambrosio and S. Luke. Hybrid agent-based and discrete event simulation in mason. In *2023 Annual Modeling and Simulation Conference (ANNSIM)*, pages 84–94. IEEE, 2023.

[22] G. D'Ambrosio and S. Luke. Hybrid agent-based and discrete event simulation in mason. In *2023 Annual Modeling and Simulation Conference (ANNSIM)*, pages 84–94. IEEE, 2023.

[23] I. M. Erma Suryani, Rully Agus Hendrawan and R. Indraswari. A simulation model to improve the value of rice supply chain (a case study in east java – indonesia). *Journal of Simulation*, 16(4):392–414, 2022.

[24] B. Fahimnia, C. S. Tang, H. Davarzani, and J. Sarkis. Quantitative models for managing supply chain risks: A review. *European journal of operational research*, 247(1):1–15, 2015.

[25] M. Galeotti. Introduction: Global crime today. *Global Crime*, 6(1):1–7, 2004.

[26] A. Goswami, A. Baveja, X. Ding, B. Melamed, and F. Roberts. An integrated framework for modeling pharmaceutical supply chains with disruptions and risk mitigation. *Annals of Operations Research*, pages 1–26, 2024.

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

[28] S. C. Graves and S. P. Willems. Supply chain design: safety stock placement and supply chain configuration. *Handbooks in Operations Research and Management Science*, 11:95–132, 2003.

[29] B. Hammi, S. Zeadally, and J. Nebhen. Security threats, countermeasures, and challenges of digital supply chains. *ACM Computing Surveys*, 55(14s):1–40, 2023.

[30] K. B. Hendricks, V. R. Singhal, and R. Zhang. The effect of operational slack, diversification, and vertical relatedness on the stock market reaction to supply chain disruptions. *Journal of Operations Management*, 27(3):233–246, 2009.

[31] G. A. Jastrebski and D. V. Arnold. Improving evolution strategies through active covariance matrix adaptation. In *2006 IEEE International Conference on Evolutionary Computation*, pages 2814–2821. IEEE, 2006.

[32] M. Jażdżewska-Gutta and P. Borkowski. As strong as the weakest link. transport and supply chain security. *Transport Reviews*, pages 1–22, 2022.

[33] J. P. Kleijnen. Supply chain simulation tools and techniques: a survey. *International Journal of Simulation and Process Modelling*, 1(1-2):82–89, 2005.

[34] P. R. Kleindorfer and G. H. Saad. Managing disruption risks in supply chains. *Production and Operations Management*, 14(1):53–68, 2005.

[35] R. Koh, E. W. Schuster, I. Chackrabarti, and A. Bellman. Securing the pharmaceutical supply chain. Technical report, Auto-ID Labs, Massachusetts Institute of Technology, 400 Technology Sq., Building NE46, 6th Floor, ambridge, MA 02139, 2003.

[36] S. Lechler, A. Canzaniello, B. Roßmann, A. Heiko, and E. Hartmann. Real-time data processing in supply chain management: revealing the uncertainty dilemma. *International Journal of Physical Distribution & Logistics Management*, 2019.

[37] S. Luke. *Essentials of Metaheuristics.* Lulu, second edition, 2013. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.

[38] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.

[39] T. Männistö, J. Hintsa, and L. Urciuoli. Supply chain crime–taxonomy development and empirical validation. *International Journal of Shipping and Transport Logistics*, 6(3):238–256, 2014.

[40] I. Manuj, J. T. Mentzer, and M. R. Bowers. Improving the rigor of discrete-event simulation in logistics and supply chain research. *International Journal of Physical Distribution & Logistics Management*, 39(3):172–201, 2009.

[41] M. Milgate. Supply chain complexity and delivery performance: an international exploratory study. *Supply Chain Management: An International Journal*, 2001.

[42] K. J. Mizgier, M. P. Jüttner, and S. M. Wagner. Bottleneck identification in supply chain networks. *International Journal of Production Research*, 51(5):1477–1490, 2013.

[43] K. Moons, G. Waeyenbergh, and L. Pintelon. Measuring the logistics performance of internal hospital supply chains–a literature study. *Omega*, 82:205–217, 2019.

[44] C. C. Nobo and R. D. Pfeffer. Natural disasters and crime: Criminological lessons from hurricane katrina. In *Climate change from a criminological perspective*, pages 173–183. Springer, 2012.

[45] J. B. Oliveira, M. Jin, R. S. Lima, J. E. Kobza, and J. A. B. Montevechi. The role of simulation and optimization methods in supply chain risk management: Performance and review standpoints. *Simulation Modelling Practice and Theory*, 92:17–44, 2019.

[46] M. Reed, J. F. Miller, and P. Popick. Supply chain attack patterns: Framework and catalog. *Office of the Deputy Assistant Secretary of Defense for Systems Engineering*, 2, 2014.

[47] M. Rizou, I. M. Galanakis, T. M. Aldawoud, and C. M. Galanakis. Safety of foods, food supply chain and environment within the COVID-19 pandemic. *Trends in Food Science & Technology*, 102:293–299, 2020.

[48] L. Roberts. *Countermeasures for Preventing Malicious Infiltration on the Information Technology Supply Chain*. PhD thesis, Purdue University, 2023.

[49] E. O. Scott and S. Luke. Ecj at 20: toward a general metaheuristics toolkit. In *Proceedings of the genetic and evolutionary computation conference companion*, pages 1391–1398, 2019.

[50] S. Serdarasan. A review of supply chain complexity drivers. *Computers & Industrial Engineering*, 66(3):533–540, 2013.

[51] E. Settanni, T. S. Harrington, and J. S. Srai. Pharmaceutical supply chain models: A synthesis from a systems view of operations research. *Operations Research Perspectives*, 4:74–95, 2017.

[52] N. Shah. Pharmaceutical supply chains: key issues and strategies for optimisation. *Computers & chemical engineering*, 28(6-7):929–941, 2004.

[53] A. C. Silva, C. M. Marques, and J. P. de Sousa. A simulation approach for the design of more sustainable and resilient supply chains in the pharmaceutical industry. *Sustainability*, 15(9):7254, 2023.

[54] L. V. Snyder, Z. Atan, P. Peng, Y. Rong, A. J. Schmitt, and B. Sinsoysal. Or/ms models for supply chain disruptions: A review. *Iie Transactions*, 48(2):89–109, 2016.

[55] K. Sullivan, M. Coletti, and S. Luke. Geomason: Geospatial support for mason. *Dept. of Computer Science, George Mason Univ., Fairfax, VA*, 2010.

[56] A. A. Tako and S. Robinson. The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Decision Support Systems*, 52(4):802–815, 2012.

[57] C. S. Tang. Robust strategies for mitigating supply chain disruptions. *International Journal of Logistics: Research and Applications*, 9(1):33–45, 2006.

[58] S. Toba, M. Tomasini, and Y. H. Yang. Supply chain management in hospital: a case study. *California Journal of Operations Management*, 6(1):49–55, 2008.

[59] L. Urciuoli, T. Männistö, J. Hintsa, and T. Khan. Supply chain cyber security—potential threats. *Information & Security: An International Journal*, 29(1), 2013.

[60] D. S. Utomo, B. S. Onggo, S. Eldridge, A. R. Daud, and S. Tejaningsih. Eliciting agents' behaviour using scenario-based questionnaire in agent-based dairy supply chain simulation. *Journal of Simulation*, 16(1):58–72, 2022.

[61] J. Wang, H. Zhou, X. Sun, and Y. Yuan. A novel supply chain network evolving model under random and targeted disruptions. *Chaos, Solitons & Fractals*, 170:113371, 2023.

[62] D. E. Whitney, J. Luo, and D. A. Heller. The benefits and constraints of temporary sourcing diversification in supply chain disruption and recovery. *Journal of Purchasing and Supply management*, 20(4):238–250, 2014.

[63] T. Wu, J. Blackhurst, and P. O'Grady. Methodology for supply chain disruption analysis. *International Journal of Production Research*, 45(7):1665–1682, 2007.

[64] Y. R. Wu, L. H. Huatuco, G. Frizelle, and J. Smart. A method for analysing operational complexity in supply chains. *Journal of the Operational Research Society*, 64(5):654–667, 2013.

[65] J. Xu, J. Zhuang, and Z. Liu. Modeling and mitigating the effects of supply chain disruption in a defender–attacker game. *Annals of Operations Research*, 236(1):255–270, 2016.

[66] D. Yang, M. Tang, and Y. Ni. Robustness of automotive supply chain networks based on complex network analysis. *Electronic Commerce Research*, pages 1–28, 2024.

[67] C. W. Zobel and L. Khansa. Characterizing multi-event disaster resilience. *Computers & Operations Research*, 42:83–94, 2014.

[68] S. Zokaee, A. Jabbarzadeh, B. Fahimnia, and S. J. Sadjadi. Robust supply chain network design: an optimization model with real world application. *Annals of Operations Research*, 257(1):15–44, 2017.