Reinforcement Learning and Scenario-Based Order for Modeling Enterprise Resilience of Maritime Container Ports

A

Dissertation

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Davis C. Loose

May 2023

APPROVAL SHEET

This Dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Author: Davis C. Loose

Advisor: James H. Lambert

Committee Member: Julianne Quinn

Committee Member: Sara L. Riggs

Committee Member: Majid Shafiee-Jood

Committee Member: Rafael C. Alvarado

Committee Member: Joi Y. Williams

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

Abstract

Global logistics systems met a crisis from the pandemic, diminished workforce, supply reductions, and demand surges. Maritime ports in particular are vulnerable to these disruptions. There is a need for methods to address system resilience. This dissertation introduces the cyber-physical systems requirements methodology (CPSRM), an approach for developing resilience of cyber-physical systems to disruptions. The CPSRM and associated tools are demonstrated in four parts on a maritime port and surrounding region as follows. First, it describes an approach to the development of a system specification as well as a hazard and gap analysis of resilience techniques. Second, it describes a mathematical simulation to account for key factors, focusing on bottlenecks in the supply chain. Third, it adapts reinforcement learning to understand and control these processes in scenarios of disruption. Fourth, it describes how to manage the disruption of system orders by the scenarios. The CPSRM improves on existing methods by incorporating particular tools from cybersecurity and risk analysis; a) red and blue team exercises for the negotiation of system requirements and b) quantification of risk as the degree of order disruption. The approach is of interest across topics of systems engineering, particularly for requirements elicitation, gap analysis, modeling and simulation, reinforcement learning, performance evaluation, and risk analysis. Practitioners will benefit by using the CPSRM to design and evaluate alternatives for system resilience.

Keywords: Systems engineering, risk analysis, enterprise systems, scenario-based preferences, modeling and simulation, hazard analysis, augmented intelligence, MuZero

Acknowledgements

I am grateful to the many people, organizations, and institutions that have supported me during my time at the University of Virginia, without whom this dissertation would not be possible. I would like to express my appreciation for my advisor James Lambert, who has been an invaluable source of unwavering support, advice, and knowledge. His experience and insight were critical for my development in academia and I am grateful for his mentorship. I am grateful to my committee – Professor Julianne Quinn, Professor Rafael Alvarado, Professor Sara Riggs, Professor Majid Shafiee-Jood, and Professor Joi Williams for their time, expertise, support, and guidance during the development of this dissertation. I am grateful to Tom Polmateer at UVA for his support and encouragement. I am grateful to Daniel Hendrickson at the Port of Virginia for his keen insight and feedback. I would like to thank Mark Manasco at the Commonwealth Center for Advanced Logistics Systems for his insights. I am grateful to Professor Cody Fleming of Iowa State University for teaching and encouraging me early in my time at UVA. I am grateful to Professor Peter Beling and Professor Stephen Adams of Virginia Tech for their advice and guidance during my time at UVA. I would like to thank the faculty and staff in the School of Engineering and Applied Sciences as well as the Engineering Systems and Environment Department for supporting graduate students. I would like to thank my peers and lab mates who have supported my work throughout my career. I am grateful to my friends and family for supporting me during my time at UVA.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Overview

This chapter provides an overview of the purpose and scope of this dissertation as well as an overview of the organization of the document. Section 1.2 describes the motivation and philosophical approach of this dissertation and introduces the new methods and models. Section 1.3 describes the purpose and scope of the dissertation derived from the motivation. Section 1.4 describes the contributions to the theory, methodology, and philosophy of systems engineering. Section 1.5 describes the structure and organization of the dissertation.

1.2 Motivation

Systems engineering management is critical in the design, implementation, and maintenance of large, complex, and interconnected systems. Systems engineering management expands beyond project management, focusing on technical and engineering components of the system with an emphasis on technical planning (SeBOK 2023). Technical planning begins at the design phase but is performed throughout the system lifecycle. Technical planning requires inputs from multiple disciplines, integration of risk and resilience management, and proper control of schedules, costs, and broader impacts to society. Further, there is a need to develop systems design processes that can incorporate risk, resilience, security, and equity at scale (United Nations CEB 2022). This process becomes increasingly difficult with the proliferation of cyber-physical systems (CPS). CPS are systems that utilize computing hardware, software, and networks that interact with physical processes and activities in the real world (ISO 2023). The cyber layer (also called the intelligent layer) of CPS introduces new solutions and opportunities for improving systems, but also introduces new sources of risk (Khalil *et al*. 2023).

This dissertation introduces a novel framework and approach for the design of cyber-physical systems, the cyber-physical systems requirements methodology (CPSRM) as described in Figure 1.1. The CPSRM is executed across six steps and utilizes four teams of experts. The CPSRM is used throughout the system lifecycle, assisting with the development of system requirements that address resilience. The CPSRM aligns with best practices for resilience analysis of CPS, incorporating:

1. An in-depth system specification for normal operating conditions
2. An analysis of disruptive scenarios and the consequences on system priorities
3. Resilience strategies to mitigate the impacts of disruptions and for performance restoration

There are several existing frameworks for the development and analysis of CPS. However, many of these techniques lack several key activities and definitions including delineation of responsibilities and scope, active collaboration with cross functional experts, and other limitations (Cassottana *et al.* 2023). These features are essential for the sustained success of complex CPS

(SeBOK 2023). That is, many frameworks for CPS design neglect sets of critical stakeholders with different perspectives and expertise. The CPSRM expands on these best practices by incorporating regimented and specific activities for multidisciplinary teams, accessing knowledge from multiple disciplines and defining the interdependencies of each task in the analysis, per the best practices of the Systems Engineering Body of Knowledge (SeBOK 2023). The CPSRM presents an opportunity for iterative analysis that can be performed throughout the CPS lifecycle, a key activity that is lacking from many existing frameworks for resilience analysis of CPS (McDermott *et al.* 2022). Further, the CPSRM utilizes reinforcement learning as a core element of CPS design and analysis of analysis, addressing another limitation of existing CPS design methods – that the system must be able to "learn from history and be unsupervised" (Darwish and Hassanien 2018).

The CPSRM is an extension of an earlier approach for requirements development for CPS, the cyber-security requirements methodology (CSRM) (Bakirtzis *et al.* 2022, Carter *et al.* 2019). The CPSRM presented in this dissertation expands on the CSRM by including a set of risks to the CPS beyond cybersecurity and includes natural and human-caused hazards. Second, the CPSRM expands on the CSRM by including an assessment of disruptive scenarios in aggregate, analyzing the impacts different hazards have on system priorities. This dissertation considers risk to be the influence of scenarios to priorities (Lambert *et al.* 2022). Given this perspective, the inclusion of the analysis of disruptive scenarios on system priorities in the CPSRM provides greater insight into the risks to the system and their influence on various aspects of the system.

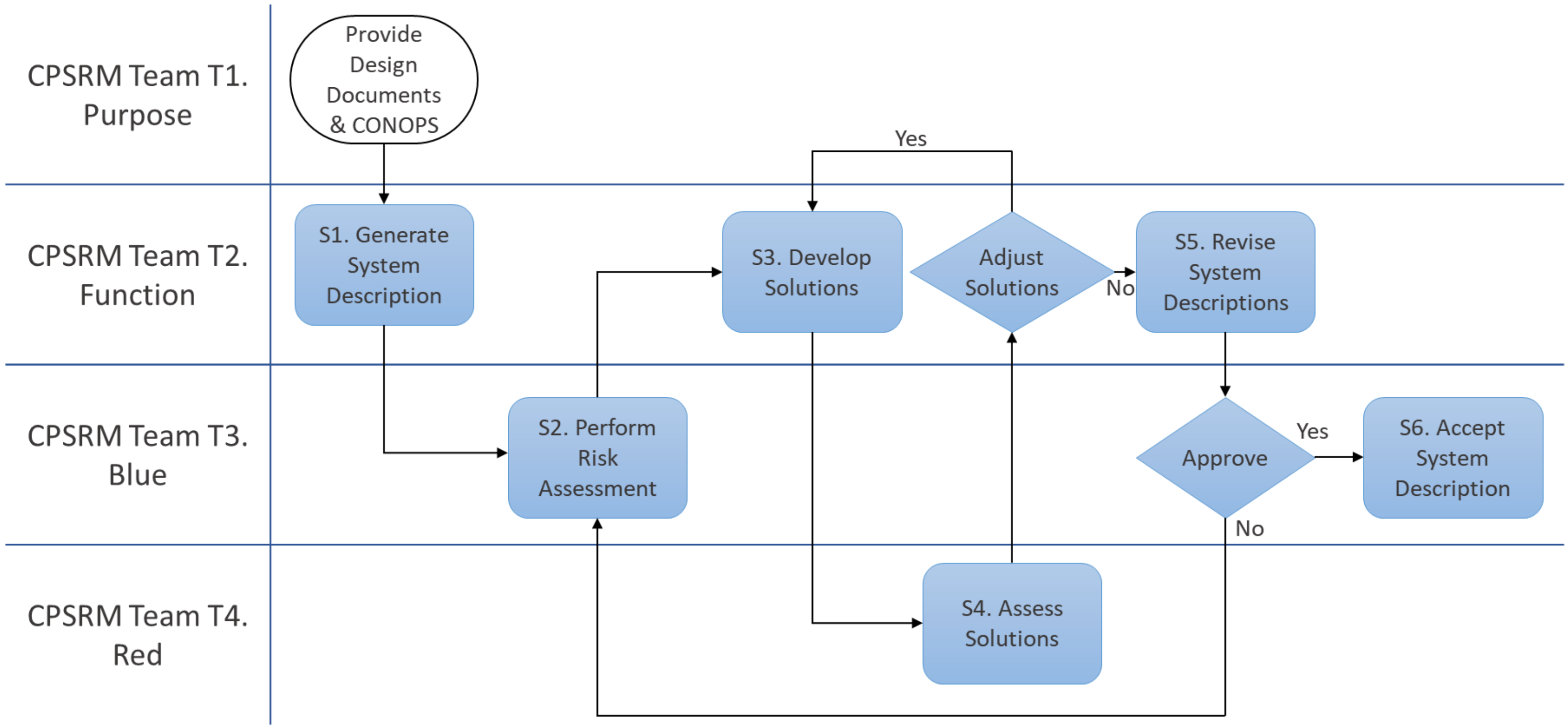

Figure 1-1. Flow chart diagram of the CPSRM including stages and teams

Managing disruptions to CPS is critical for maintaining operations during periods of disruption in large logistics systems (Eddy *et al.* 2022). Hazards such as hurricanes, flooding, pandemics, and climate change-induced disasters disrupt global supply chain networks (Wu *et al.*, 2022, 2021, Loose *et al.* 2021). Further, resources are limited and balancing the costs and benefits of resilience measures is a critical decision point for the design and maintenance of CPS (Bonato *et al*. 2021). This dissertation demonstrates the CPSRM on operations at a large maritime container port – the Port of Virginia in Norfolk, Virginia – and a surrounding 500-mile radius. This demonstration explores how disruptions influence the design of systems, how simulations can be used to test new resilience measures, and how reinforcement learning can be applied to controls problems in CPS to manage operations. These results are analyzed to understand how disruptive scenarios influence system priorities. This system is of particular interest for several reasons. First, it is a priority of the port to increase automation and reliance on cyber-systems to address supply chain disruptions and constraints – therefore the port needs a reliable way to develop new requirements for enterprise systems. Second, improving the performance of ports and maritime logistics at large presents an opportunity to increase equality and equity, improving future living conditions at a global scale. This demonstration presents an opportunity to implement sustainability and fairness at ports, explore the ethics of artificial intelligence in CPS, and explore the future of systems engineering as a discipline.

Growing interconnection across global markets has increased the importance of reliable logistics systems (Kamalahmadi *et al.* 2022, Chopra and Sodhi 2014). There is a relationship between social equity along racial, ethnic, and gender lines, and supply chain reliability. It is critical when evaluating supply chains to consider geographic, socioeconomic, racial, and gender diversity in decision making (CSIS 2022). The COVID-19 pandemic revealed weaknesses in global supply networks. In the United States alone, total freight movement decreased by nearly 22% in the wake of supply shocks caused by COVID-19 (US DOT 2022). This shock exacerbated inequities in the racial wealth gap (White House 2021).

Ships deliver over 80% of world trade across dozens of sectors including food, energy, and medicine. This 80% figure is higher for developing countries, driving the need for building resilience to disruptions that inhibit supply chains. The COVID-19 pandemic exposed limitations to the maritime shipping industry as prices rose to historic levels. Figure 1-2 describes the change in container price index before and during the pandemic.

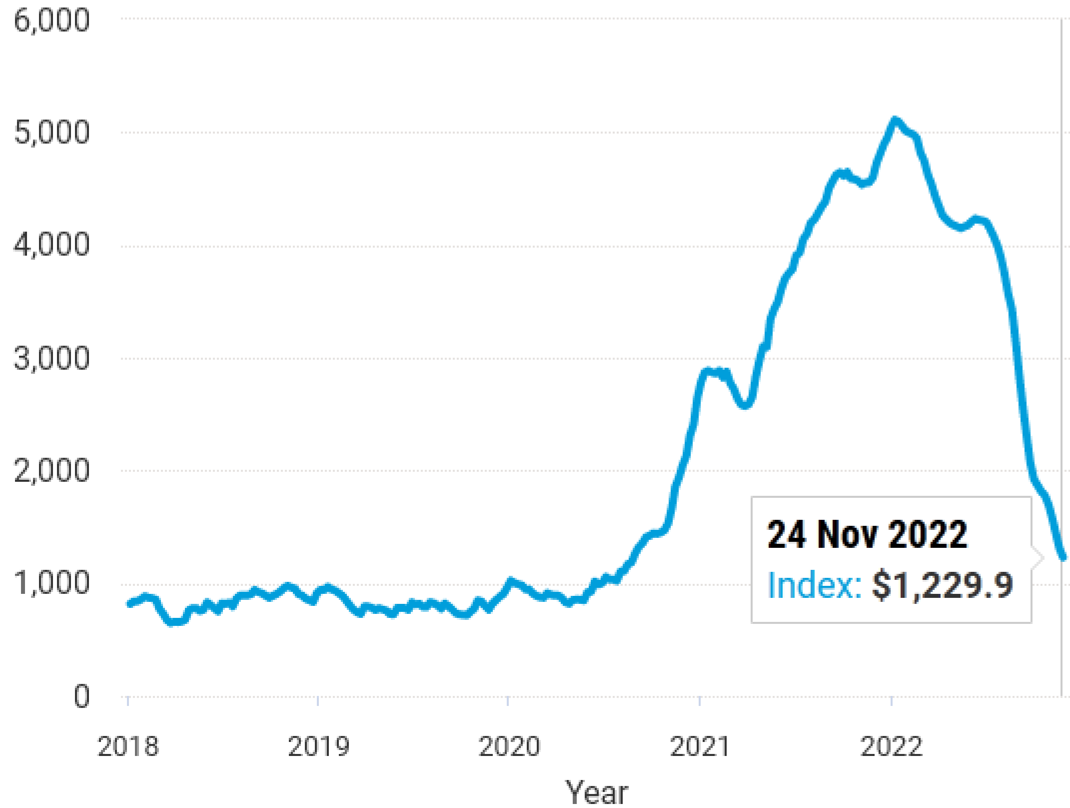

Figure 1-2. Depiction of the rapid rise and fall of shipping costs in the period before and during the COVID-19 pandemic. The COVID-19 pandemic caused supply chain shocks that increased the costs of containerized and bulk shipping. Though prices have lowered from their peak in January 2022, shipping costs remain higher than pre-pandemic levels. Adapted from (UNCTAD 2022).

As such, the shipping industry is being asked to invest in new technologies and processes that allow supply chains to withstand disruption. Further, this technology should be sustainable with respect to the environment and focus on equity (UN News 2022). Major maritime ports are often constructed in locations already facing poor socioeconomic conditions, introducing compounding harms due to the emissions of the port (Hendrickson 2023). The United Nations

Conference on Trade and Development (UNCTAD) Secretary General Rebecca Grynspan said on the modernization and automation of ports:

> "We must change course and we must do it now. To prepare for the future, we need shipping and supply chains to be more efficient, more resilient, and far greener. The world should prepare for unpredictable future with volatile shipping costs." (Grynspan 2022).

Volatility in shipping rates impacted low- and middle-income countries more than high-income nations. Figure 1-3 describes the price changes of food items across three types of nations: high, medium, and low-income. Low-income countries face a 0.82% increase in grain prices compared to 0.47% increase for high income countries during the shipping shocks due to the COVID-19 pandemic. Further, midline income nations face 1.27% total increase in food prices compared to a 0.81% increase in food prices in high-income nations. While shipping costs rose and have begun to fall in the aftermath of the pandemic, the price of goods remains high due to the shock (UNCTAD 2022).

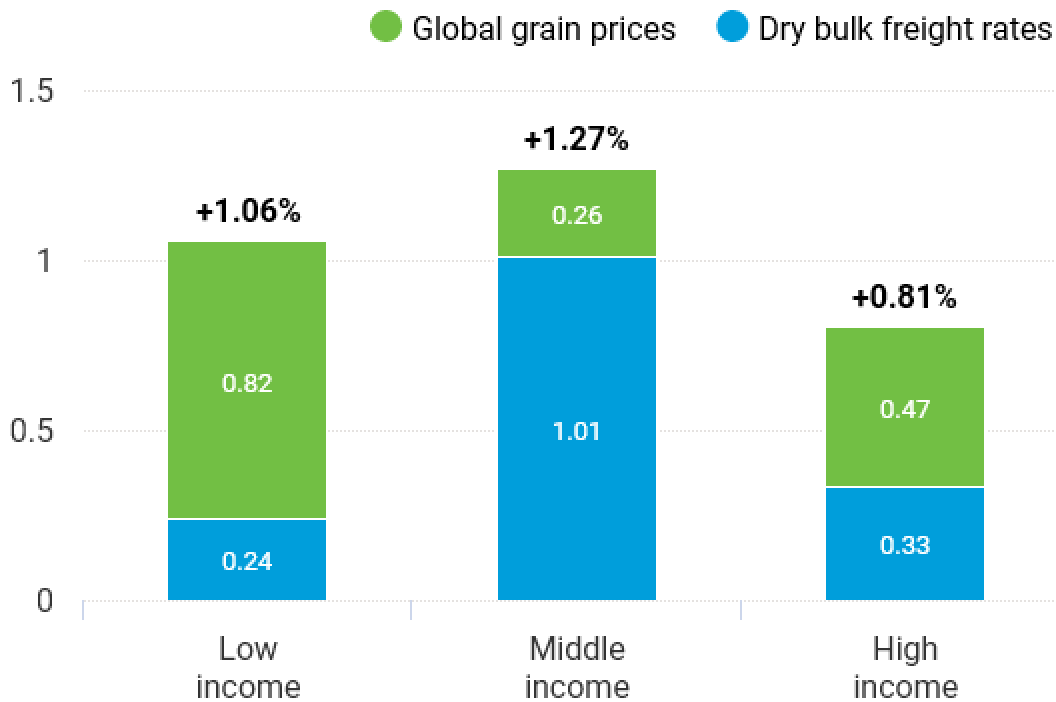

Figure 1-3. Long term impact of global fluctuations in shipping costs on food prices in low, middle, and high-income nations. The supply chain shocks of COVID-19 have not impacted all nations equally. Nations with developing and emerging economies face greater price increases for food than wealthy nations. Adapted from (UNCTAD 2022).

These price increases are not expected to return to 2019 levels. In response to this, the UNCTAD 2022 report on the state of maritime trade and transportations calls for:

> "1. Governments and operators to expand and upgrade port infrastructure and land transport connections, and accelerate trade facilitation reforms, especially digitalization.
>
> 2. Port operators and shipping companies to invest in increasing storage facilities and reducing equipment shortages.

> 3. Shipping companies to invest in sustainable shipping and deploy the necessary ship-carrying capacity." (UNCTAD 2022).

A primary focus of the 2030 United Nations Agenda for Sustainable Development is the improvement of port performance, connectivity, and automation (UNDESA 2022). A long-term lack of investment has affected port performance across the globe, weakening markets and economic growth. African, Latin American, and Caribbean ports lost more than 10% of direct shipping connections in the wake of the pandemic. Contrast this with ports in India, China, and Norway, which focused on modernization and increasing connectivity throughout the pandemic and realized slight growth (UNCTAD 2022). Affordable, actionable techniques for improving port performance and container handling are necessary steps for reducing this disparity. Specifically, the UN calls for:

> "… [Supporting] developing countries to improve port performance and productivity, including by upgrading port capacity and strengthening regional transport connections." (UNDESA 2022).

Container ports are critical elements in the development and growth of emerging economies. Increasing the attractiveness of these ports to industry and governments is an opportunity to reduce inequalities between these nations and the existing powers. Further, many developing nations have borne the brunt of the effects of climate change. Policies and technologies that improve sustainability may be expensive, and new policies limiting emissions may prevent developing economies from keeping pace (UNEP 2021). It is critical for the international community to ensure that climate and disruption mitigation efforts do not severely inhibit new growth in developing nations. Nichola Peltier-Thiberge, the Global Practice Director of Transport for The World Bank, describes how port investments can stimulate growth in emerging economies:

> "In many cases, the development of high-quality container port infrastructure, operated efficiently, has been a prerequisite to successful export-led growth strategies. It can facilitate investment in production and distribution systems, supporting the expansion of manufacturing and logistics, creating employment, and raising income levels." (WBG 2022)

Managing logistics through maritime ports is critical for reducing inequalities in already wealthy nations. In the United States, as of November 2022, 68% of the total wealth of citizens is owned by the top 10% of earners. Further, the bottom half of all earners own only 3.3% of the total wealth of the United States (Statista 2023, US Census 2022). Since 1993, the income inequality between low and high earners has steadily increased, a total of an 8.8% in the Gini index (a measure which reflects the amount that two incomes differ when compared to the mean income). Income inequality had been steadily falling since 2011, but the COVID-19 pandemic reversed this trend, with income inequality growing by 1.2% from 2020 to 2021 (Semega and Killar 2022).

Maritime ports present an opportunity to reduce this inequality. As of 2018, US ports supported over 30.8 million jobs and fueled 26% of US GDP. Seaports are often located near or in metropolitan areas and the decisions made by ports may impact millions of individuals, especially with respect to sustainability and the environment. These communities are often low or middle-income and have limited input on port activities. Thus, there is a need for innovation, in particular in advanced analytics and automation to improve sustainability, resilience, and emissions (ASCE 2022). Per the American Society of Civil Engineers:

> "Advanced analytics… aid ports in becoming more resilient as predictive approaches driven by machine learning ensure flexible,

> responsive, and adaptive management amid highly complex and dynamic scenarios." (ASCE 2022)

This need is reflected in national policy. Over $2.7 billion is invested in funding focused on improving American ports and waterways for fiscal years 2022 and 2023 alone. Much of this funding targets environmental justice projects – infrastructure improvement plans that improve quality of life in disadvantaged communities (The White House 2022). Beyond this, general improvements to port infrastructure to bolster resilience to disruption is a primary national concern.

> "The top economic priority of the [United States] White House is fighting inflation… through managing port disruptions and [easing] bottlenecks. [The goal] is not only to get through the current bottleneck, but to address the longstanding weaknesses in our supply chain that this pandemic exposed."
>
> *President Joseph Biden, June 2022*

The goal of not only improving current performance but increasing resilience moving forward requires careful, clear, regimented systems engineering design. This design will require inputs from diverse stakeholders to meet the many myriad objectives of port improvements, including increased efficiency, reduced emissions, and social equity. The CPSRM presented in this dissertation addresses these design needs, providing an approach for identifying requirements, addressing resilience, testing new analytic techniques, and understanding disruptions.

1.3 Purpose and Scope

The purpose of this dissertation is to describe and provide a demonstration of the CPSRM, a philosophical methodology for the design and analysis of resilience in cyber-physical systems. The methodology advances systems design by incorporating aspects of cyber-security, systems planning, and risk analysis into a single framework for requirements gathering. This approach enables organizations to address the systemic and global issues that impact logistics systems, vulnerable communities, and sustainability.

This methodology can be applied to any CPS, and preliminary versions have been used on defense and medical systems with an increased focus on cyber threats (Beling, Loose *et al.* 2021, 2020). This dissertation expands the CPSRM and is demonstrated on a maritime port and surrounding industrial region to address the motivating factors in the previous section. First, the CPSRM is used to identify areas of opportunity in port processes, generate a description of this process, and analyze potential disruptors, including a gap analysis. Second, the CPSRM uses the description to create a simulation model, providing a tool on which new resilience techniques can be tested and new ideas explored. Third, this dissertation implements the MuZero reinforcement learning algorithm to a port logistics system as a method for improving system resilience. Finally, the system is analyzed at a macro-level using a mathematical framework to identify how system priorities change when exposed to disruption.

1.4 Contributions

This dissertation describes an approach for developing requirements for the design of resilient cyber-physical systems. The development of this approach advances systems engineering as a discipline in six respects.

Contribution 1. Development of the CPSRM. Developed the cyber-physical systems requirements methodology (CPSRM) as an approach for identifying and collecting requirements for cyber-physical systems to improve system resilience. The CPSRM algins with the standard framework for the design and analysis of CPS, and expands on this standard by incorporating elements from multiple disciplines including cybersecurity and risk. The CPSRM also advances CPS design paradigms by incorporating cross and multifunctional teams throughout. The CPSRM is demonstrated on the Port of Virginia – specifically the container handling process – and the surrounding regions (Beling, Loose *et al.* 2021, 2020).

Contribution 2. System specification. Developed a system specification process for considering the goals, missions, and other factors influencing the CPS. The system specification builds on work from the computer science discipline, generating the process for systems engineering. The system specification includes disruptive scenarios and their outcomes as a factor in the system design. The system specification is demonstrated on the Port of Virginia, specifically the dynamics and performance of a container stacking block for ten stacks and five tiers, handling more than 175 containers across a week of operation (Loose *et al.* 2023(b), Hamdy, Loose *et al.* 2022).

Contribution 3. Gap analysis. Developed a hazard and gap analysis methodology to address system resilience to natural and human-caused hazards that impact the operations of the

CPS. The hazard and gap analyses are inductive methods for enumerating known threats to the CPS, outlining the potential impacts to performance metrics due to disruption. The hazards are described for the Commonwealth of Virginia and the Hampton Roads region containing the Port of Virginia, addressing the risk profiles of each hazard. The gap analysis identifies new resilience techniques to be tested with the CPSRM for potential inclusion in the final set of resilience requirements. The gap analysis identifies a reinforcement learning control algorithm as a hazard mitigation method for reducing the harms of disruption to the container stacking problem (VDEM, Loose *et al.* 2023, Hill, Loose *et al* 2023).

Contribution 4. System simulation. Developed a simulation as a part of a solution for mitigating the impacts of disruptive scenarios. The simulation utilizes the system specification, hazard analysis, and gap analysis in its design and serves as a test bed for implementing new resilience techniques in the system. The simulation is used to test the performance of the reinforcement learning algorithm, as well as examine how system design decisions may impact operations. The simulation of a container stacking block is used to compare performance across key metrics such as the number of container moves, and is the test bed for the MuZero control algorithm (Loose *et al.* 2023(b), Costello, Loose *et al* 2023).

Contribution 5. Reinforcement learning. Developed and trained a reinforcement learning algorithm as a method for increasing system resilience. This dissertation is the first to apply the MuZero reinforcement learning algorithm to the container stacking problem as a trained model to control the simulation of the container stacking block. The model is used for planning and operational improvements to the container stacking process. The algorithm serves as a resilience technique, improving port performance in disrupted and non-disrupted scenarios. MuZero achieves similar performance to the Port of Virginia in terms of touches per container under typical conditions and demonstrates how performance can be improved during periods of disruption (Loose *et al.* 2023(b)).

Contribution 6. Mathematical framework. Developed and applied a mathematical framework for assessing how disruptive scenarios influence system priorities. This dissertation expands on prior work by analyzing how stakeholder disposition and risk tolerance impacts the order of priorities. The framework is demonstrated across three cases in a region near the Port of Virginia, and focuses on supply chain and logistics resilience (Loose *et al.* 2023(a), 2022(a), 2022(c)). These contributions are described in this dissertation as outlined in Table 1-1.

Table 1-1. Relationship between contributions to systems engineering and the chapters in which the contributions are described

Contribution	Related Chapters
I	Ch.3, Ch.4, Ch.5, Ch.6, Ch.7
II	Ch.3, Ch.4
III	Ch.3, Ch.4
IV	Ch.5
V	Ch.6
VI	Ch.3, Ch.4, Ch.7

Chapter 3 describes the CPSRM including its purpose, use cases, and an overview of the constituent elements focusing on how it differs from and improves upon existing systems engineering and systems design methodologies. Chapter 4 describes the first two stages of the CPSRM, including the process for conducting a system specification, outlining the potential disruptors of the system, and providing a gap analysis of resilient solutions. Chapter 5 describes the development of a simulation of a particular process of the maritime port system, specifically developed for designing and testing the performance of a control algorithm. Chapter 6 describes an example control algorithm, a novel use of a reinforcement learning algorithm to automatically control processes, improve system resilience, and investigate how design changes may impact performance. Chapter 7 describes an extension of a mathematical framework used to analyze systems, focusing on how emergent conditions impact the prioritization of resilience requirements and a quantification of disruption, as well as exploring the sensitivity of the framework.

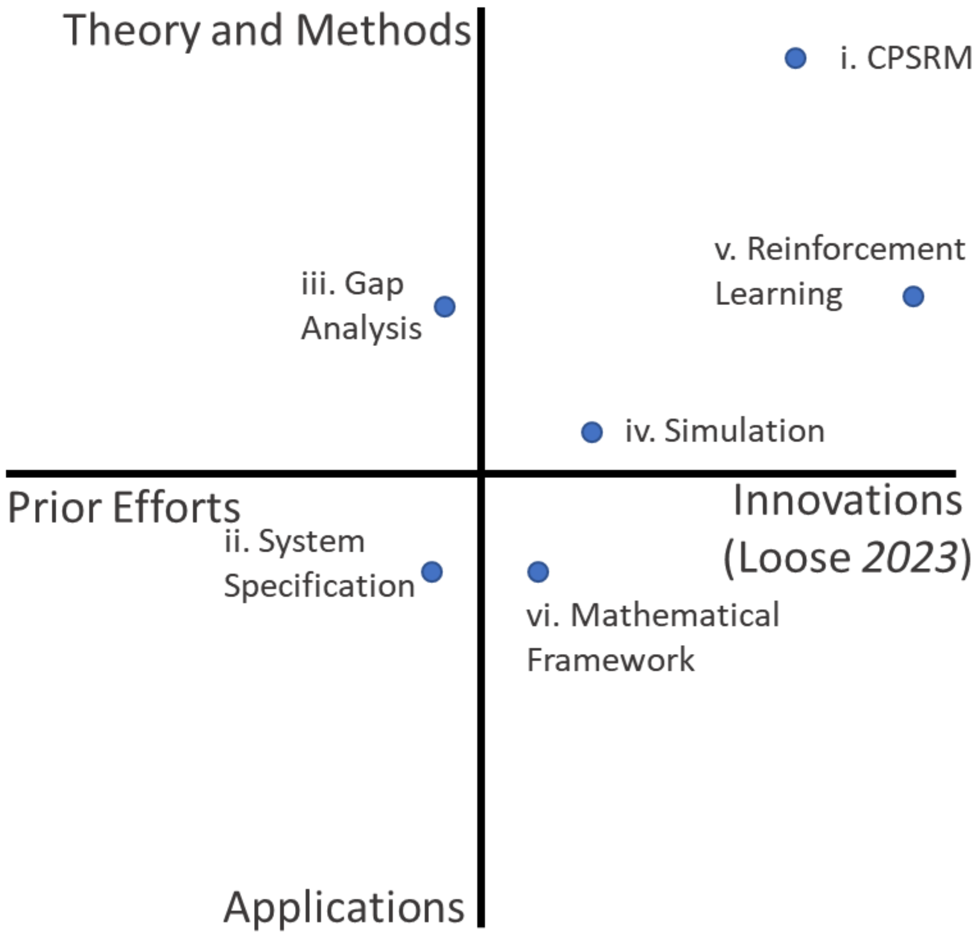

Figure 1-4. A graph exploring the degree to which each contribution is an innovation and the degree to which each contribution is applicable in a general or specific use case

These contributions are accepted, published, or under review in three journal papers (Loose *et al.* 2023, Eddy, Loose *et al.* 2023, Andrews, Loose *et al.* 2023), four conference papers (Lambert, Loose *et al.* 2022, Eddy, Loose *et al.* 2022, Hamdy, Loose *et al.* 2022(b), Loose *et al.* 2023(a)), four presentations to conferences and government (Loose *et al.* 2022(b), Loose *et al*. 2022(c), Loose *et al.* 2021, Beling, Loose *et al.* 2020), and three published reports (Lambert, Loose *et al.* 2022, FEMA, Loose *et al.* 2023, Beling, Loose *et al.* 2020). A further three papers are accepted, in revision, or submitted to conferences or journals (Loose *et al.* 2023(a), 2023(b), Hill, Loose *et al.* 2023).

Table 1-2. Research and its venue of publication or presentation, including the distribution across chapters

Presentation or Publication	Related Chapter
SERC, Beling, Loose *et al.* 2020	Ch.3
SERC, Beling, Loose *et al.* 2021	Ch.3, Ch.5
CESUN, Loose *et al.* 2021	Ch.7
CPDC, Lambert, Loose *et al.* 2022	Ch.7
CPDC, Loose *et al*. 2022(a)	Ch.7
IEEE SYSCON, Loose *et al.* 2022(b)	Ch.7
IEEE SYSCON, Eddy, Loose *et al.* 2022	Ch.3
IEEE SYSCON, Loose *et al.* 2022(c)	Ch.7
IEEE SIEDS, Hamdy, Loose *et al.* 2022	Ch.4, Ch.5, Ch.6
Wiley journal *Risk Analysis*, Andrews, Loose *et al.* 2022	Ch.7
Submitted Wiley journal *Risk Analysis*, Loose *et al.* 2023(a)	Ch.7
Wiley journal *Systems Engineering*, Eddy, Loose *et al.* 2023	Ch.5
VDEM HIRA, VDEM, Loose *et al.* 2023	Ch.4
IEEE SIEDS, Costello, Loose *et al*. 2023	Ch.5, Ch.6
Working Paper, Hill, Loose *et al.* 2023	Ch.4, Ch.7
Submitted IEEE CoDIT, Loose *et al.* 2023(b)	Ch.5, Ch.6

1.5 Organization

Chapter 2 provides the history of and need for this work based on relevant literature from the systems engineering, cyber-physical systems, logistics, machine learning, hazard analysis, and other domains. Chapter 3 introduces and explains the CPSRM in detail. Chapter 4 outlines the system specification and gap analysis for the demonstration on the maritime port. Chapter 5 describes the development of the system simulation including, inputs, outputs, motivating issues, and other key parameters. Chapter 6 describes the implementation and training of a reinforcement learning algorithm for controlling the target system. Chapter 7 describes an extension of a mathematical framework for analyzing disruptions of system priorities. Chapter 8 summarizes the conclusions of this dissertation, including contributions, plans for future work, and broader implications. These are followed by a References section and Appendices with supplementary data. Figure 1-5 describes the breakdown of chapters into categories.

Introduction

Chapter 1: Introduction

Chapter 2: Literature review

Development of Methods

Chapter 3: Methods

Method Applications

Chapter 4: System specification and gap analysis

Chapter 5: Development of system simulation

Chapter 6: Integration of reinforcement learning for process control and resilience

Chapter 7: Mathematical framework for analysis of disruptions to system orders

Conclusions

Chapter 8: Summary and conclusion

Back Matter

References

Appendices

Figure 1-5. Structure and overview of dissertation chapters

Chapter 2: Literature review

2.1 Overview

This chapter describes the history and leading research for each of the chapters in this dissertation. Section 2.2 describes the state-of-the-art for cyber physical system design and analysis. Section 2.3 describes the process for developing a system specification. Section 2.4 describes the process for completing a hazard and gap analysis. Section 2.5 describes the use of simulation for resilient design. Section 2.6 describes the use of reinforcement learning for controls and designing resilience. Chapter 2.7 describes the use of mathematical frameworks for understanding disruption to system order.

2.2 Models and methods for resilient design of cyber physical systems

The CPSRM described in this dissertation is an extension of the cyber security requirements methodology (CSRM) (Carter *et al.* 2019). The CSRM focuses on cyber vulnerabilities in CPS such as code weaknesses and other digital exploitations. Design changes become more expensive to implement as CPS grow and mature. As such, it is advantageous to implement resilience techniques early in the design process. *Resilience techniques* are tools, processes, or technologies that system managers can implement to reduce the impact of disruption (Sgobbu and Codara 2022, Yang and Hsu 2018). The CSRM is executed during the requirements gathering phase of systems design, enabling stakeholders to analyze the target system and select resilience techniques. Four teams coordinate to analyze, prototype, and prioritize requirements for the target system. The CSRM is paired with the systems-theoretic resiliency assessment tools (STRAT), a set of methods used to accelerate the CSRM process. The CSRM has been applied to CPS such as defense and healthcare systems (Beling, Loose *et al.* 2020). The CSRM is effective for defining and applying resilience techniques to CPS, but the focus on cyber security alone is a limitation. The CPSRM expands the CSRM to include threats and hazards outside of the cyber domain. The CPSRM maintains the four-team structure of the CSRM with alterations to responsibilities and tools, such as reducing the role of the red team to focus on assessing solutions rather than developing attack patterns. The CPSRM utilizes tools similar to STRAT to collect, test, and prioritize resilience capabilities. These tools are outlined in Chapters 4, 5, 6, and 7.

There are many CPS modeling approaches, each with different focuses, benefits, and drawbacks – for example, the ISO/SAE 21434 for road vehicle cybersecurity or the NIST SP 1500-201 Framework for cyber-physical systems. Some research indicates that these various methods are different specialized arrangements of tools, models, and methods, tailored to a specific process. Table 2-1 describes where the CPSRM falls in this spectrum including the tools used for analysis across the three primary phases for CPS (Cassottana *et al.* 2023). Table 2-2 describes the relevant literature to the analysis of the CPSRM method.

Table 2-1. Relationship between CPSRM and general-purpose CPS analysis models. CPSRM elements are in bold and underlined. The inclusion of *Automation* as a resilience technique is a contribution of this dissertation. Adapted from (Cassottana *et al.* 2023).

Resilient design for CPS	Object of analysis	Tool/Model/Method
System description	Physical components **Control components** **Cyber components**	Attribute-based models: • Graph theory models • Complex network theory models • Finite-state machines • Petri nets Performance-based models: • Dynamic system models • Agent-based models • **Simulation-based models**
Disruptive scenarios	**Physical disruptions** **Cyber disruptions**	Inductive methods: • **Contingency analysis** • Event trees • Attack graphs • Bayesian networks • Markov processes Deductive methods: • Fault trees • Attack trees • Heuristic optimization methods
Resilience techniques	**Hardening components** Redundant components Component restoration Early warning systems Intrusion tolerance Authentication **Automation**	Reactive methods: • **Control theory methods** • Game theoretical models Proactive methods: • **Analytical models** • Time-series analysis • Machine learning • **Reinforcement learning**

Table 2-2. Summary of literature on the design, development, analysis, and implementation of resilience for cyber physical systems.

Aspect of Resilient Design for CPS	References
Methods for the resilient development of CPS	Martin 2020 Weilkiens 2011 Carter 2018 Zachman 1987
Resilient design of CPS to evolving and persistent threats	CISA 2022 Sgobbi and Codara 2022
Development of CPS design techniques to address societal change	Collier *et al.* 2021(b) Collier *et al.* 2020(a)
Analysis of security of CPS	Khalil *et al.* 2023 Beteto *et al.* 2022 Bakirtzis *et al.* 2018
Designing CPS with a focus on broader missions and objectives	Laarni *et al.* 2022 Goldman *et al.* 2011
Engineering resilience in CPS	Bakirtzis *et al.* 2022 Jin *et al.* 2022 Beling, Loose *et al.* 2021 Beling, Loose *et al.* 2020 Majumder *et al.* 2017 Ali and Ronaldson 2012

Methods and techniques for establishing organizational alignment in CPS design	Chowdjury and Gkioulos 2022 McDermott *et al.* 2022 UNCEB 2016
Considering complexity when analyzing or designing CPS	Henning *et al.* 2022 Liu and Li 2021

2.3 System specification and analysis

The first stage of the CPSRM is the development of a high-level system description based on a system specification and requirements gathering exercise. Systems analysis and specification has been investigated for decades, primarily in the software development domain (DeMarco 1979, IEEE CS 1998, Bødker 2021). In recent years, system specification documents have become more common for broader systems design, especially in CPS (Suhail *et al.* 2022) Systems analysts assess the needs of stakeholders to design a system that addresses organizational missions. Several methods exist for collecting specifications for development such as interviews, document analysis, prototyping, and workshopping (Tracy 2021). The CPSRM does not specify the method of system specification in the general case, but utilizes interviews and document analysis for the maritime container port case study. Table 2-3 describes the literature motivating the development of the system specification and gap analysis in the general and specific cases.

Table 2-3. Summary of literature on the development of a system specification for cyber physical systems and maritime ports

Aspect of system specifications	References
Development, maintenance, and execution of system specifications	Suhail *et al.* 2022 Tracy 2021 Verasetti 2021 Stern *et al.* 2005 Sindre and Opdahl 2005 IEEE CS 1998 Davis 1982 DeMarco 1979
Analysis of port metrics and performance	Hassan and Gurning 2020 PoV 2018
Supply chain resilience	Sadiq *et al.* 2022 Bonato *et al.* 2021(a) Helo *et al.* 2018 Chopra and Sodhi 2014
Analysis of port planning processes	Cho 2021 Chao and Lin 2019 Almutairi 2017
Port systems and equipment	PoV 2023 Virginia Port Authority 2021 Benutzer 2004

Innovations to port equipment	AAPA 2023 PoV 2022
Motivations for improvement to port processes	SHIPA Freight 2023 Statista 2023 ASCE 2022 Grynspan 2022 ISO 2022 Semega and Killar 2022 US Census 2022 WBG 2022
Requirements engineering	Kuzmanovic 2019 Hull *et al.* 2005 Van Lamsweerde 2000 Lie 1998

2.4 Hazard and gap analysis

The second stage of the CPSRM is a gap analysis of resilience capabilities. Gap analysis has been used to enhance resilience in several types of systems, including environmental systems (Thomas *et al.* 2021) and supply chains (Jensen and Orfila 2021). The objective of such an analysis is to take inventory of existing capabilities and identify ways the system can be changed and improved. That is, identifying the resilience techniques required to navigate from the "as-is" system, to the desired "to-be" system. The gap analysis techniques used in this dissertation are based on and extend methods outlined in (Mineraud *et al.* 2016). The process begins with an assessment of known risks through techniques such as risk filtering via hierarchical holographic modeling to prioritize and assess risk scenarios (Haimes *et al.* 2002). The system is then examined to determine which resilience techniques address these risk scenarios (VDEM 2018, 2023). Other similar systems and requirements are examined to identify gaps between current capabilities and state of the art capabilities (Mittal and Cane 2016). Table 2-4 describes a summary of relevant literature regarding hazard and gap analysis.

Table 2-4. Summary of literature on methods and techniques for conducting hazard and gap analyses. Also provides current literature used to execute the hazard and gap analyses in this dissertation.

Aspect of hazard or gap analysis	References
Supply chain and logistics resilience to hazards	VDEM 2023 CSIS 2022 Mclean 2020
Resilience techniques and mitigation strategies for reducing harms from hazards	DCR 2021 USCG 2019 DHS 2018 VDEM 2018
Changes to operations in presence of hazards	Wu *et al.* 2022 Thorisson *et al.* 2020 Ambrosino and Anna 2018

Systems-of-systems resilience to natural hazards	Andrade and Hulse 2022 UNEP 2021 VCC 2021
Disruptions to transportation networks	UNCTAD 2022 Andrews *et al*. 2020(b) Andrews *et al.* 2020(c)
Metrics and analytics for resilience and sustainability	The World Bank 2022 UNDESA 2022 Collier *et al*. 2021(a)
Risk modeling under threat of hazards	Bonato *et al.* 2021 Collier *et al.* 2020(d)
Assessment of compounding and hybrid hazards	Hill, Loose *et al.* 2023 Collier *et al.* 2020(e)
Gap analysis methods and frameworks	Thomas *et al.* 2021 Mineraud 2016 Mittal 2016 Jennings 2002
Simulation and system specifications for development of resilience	Zhan *et al.* 2020 Kawahara *et al.* 2009

2.5 Simulation for resilient design

The third stage of CPSRM utilizes model-based simulations to explore new resilience capabilities on the system of interest. Simulation has been used to assess resilience in logistics systems (Coelho 2022, Pascual *et al.* 2016), supply chains (Carvalho *et al.* 2019), and environmental systems (Andrade and Hulse 2022). Simulation can take many forms such as Monte Carlo simulation or through use of systems modeling languages (Rabe *et al.* 2021, Thorisson *et al.* (2) 2019, Carter *et al.* 2019).

There are many existing simulation methods, leveraging expert systems (Chou and Fang 2021, Kim *et al.* 2019), specialized software (Larsen and Pacino 2020), or general-purpose software (Brockman *et al.* 2016) used in port systems specifically. The technique used in the CPSRM and for this demonstration is a general-purpose simulator, specially tailored for maritime port operations. There are many benefits to this method, as it allows for greater flexibility in the design and fidelity of the simulation (Cho *et al.* 2021, Thorisson *et al.* 2019, Guan *et al.* 2002). Further, it creates a more stable integration with machine learning resources (Duvaud and Hainaut 2020). Table 2-5 describes a summary of literature regarding simulation and resilience for CPS in general and port operations specifically.

Table 2-5. Summary of literature on methods and techniques for the development of simulations

Aspect of simulation and resilience	**References**
Simulation and analysis of port processes	Costello *et al.* 2023 Vlahavas and Refanidis 2013 Guan *et al.* 2002

Simulation of container stacking operations	Larsen 2020 Kim *et al.* 2019 Jimena *et al.* 2016 Pascual *et al.* 2016
Simulation for supply chains	Coelho and Barbosa-Povoa 2021 Rabe *et al.* 2021 Carvalho *et al.* 2012 Schmitt *et al.* 2009
Analysis of port systems, performance, and operations	Aegis Environmental 2019 Almutairi 2016
Expert knowledge and simulation design	Wang *et al.* 2019 Chou and Fang 2018
Simulations for the development of control strategies	Powell 2022 Blum *et al.* 2021 Rei *et al.* 2008
Design and analysis of simulations	Boedker *et al.* 2021 Carey and Rossler 2020 Coad and Yourdon 1991
Advancements in simulation technologies	Marttunen and Mustajoki, 2018 Brockman *et al.* 2016

2.6 Reinforcement learning for control and resilience

The fourth and fifth stages of the CPSRM use the model-based simulation to implement and test resilience capabilities. For this dissertation, the resilience capability of interest is a planning algorithm which controls port operations to reduce delays during and after disruptive events. Control algorithms utilizing reinforcement learning are a leading method for decision support in logistics and supply chains (Yan *et al.* 2022). MuZero is a program and reinforcement learning algorithm developed by DeepMind (Schrittwieser *et al.* 2020). The program was developed to play games such as chess, shogi, go, and Atari, but the underlying algorithm is known to improve operations of systems and tasks such as robotics (Zhan *et al.* 2020) and air traffic control (Yilmaz *et al.* 2021). The algorithm combines Monte Carlo Tree Search (MCTS) with a learned model to perform various tasks, including achieving superhuman performance at chess, and other rules-based games. Planning algorithms that utilize lookahead search such as MuZero have been successful in several domains, but in particular logistics (Powell 2022, Vlahavas and Refanidis 2013). This dissertation expands on research and applies MuZero to port operations, tuning the algorithm to integrate with the simulation-based solution (Hamdy *et al.* 2022, Duvaud and Hainaut 2020). Table 2-6 describes the use of machine learning, reinforcement learning, and other methods for improving the resilience of process controls.

Table 2-6. Summary of literature on machine learning, reinforcement learning, and other techniques for improving resilience of logistics systems.

Aspect of machine and reinforcement learning	References
Development of new reinforcement learning techniques and algorithms	Long *et al.* 2023 Liu *et al.* 2022 PoLA 2022 Duvaud and Hainaut 2020 Schrittwieser *et al.* 2020 Zhang *et al.* 2017
AI for controls	Chen 2022 Oroojlooyjadid *et al.* 2022 Pireva *et al.* 2017 Hatzi *et al.* 2011
AI for transportation	Yilmaz *et al.* 2021 Abduljabbar et al. 2019 Thorisson *et al.* 2019(a)

AI for container management	Loose *et al.* 2023(b) Bosch 2022 Hamdy, Loose *et al.* 2022 Sikorra *et al.* 2021 Maldonado *et al.* 2019 Guven and Eliiyi 2014 Fotuhi *et al.* 2013 Lee *et al.* 2007
Integration of simulators and reinforcement learning techniques	Yan *et al.* 2022 Cutler *et al.* 2014
Algorithmic approaches for decision systems and CPS	Collier *et al.* 2021(b) Jung and Jazizadeh 2019 Yang and Hsu 2018
Analysis of reinforcement learning techniques	Moerland *et al.* 2023 He *et al.* 2018 Bello *et al.* 2016

2.7 Analysis of disruption of system orders

The sixth and final stage of the CPSRM is the selection of resilience techniques to include in the system design. To assist in the process, this dissertation applies a model for prioritizing resilience measures under uncertainty as an aspect of risk analysis (Loose *et al.* 2022 (a), 2023(a)). Traditional approaches consider risk to be a function of probability, consequences, and vulnerability (Bouchat and Asveld 2020, Conrow 2007). Still others define risk as the "effect of uncertainty on objectives" (Hutchins 2018). The CPSRM utilizes a complementary perspective through the use of a mathematical ranking framework in which risk is the measure of disruption of scenarios to priorities (Loose *et al.* 2023(a), Lambert, Loose *et al.* 2022). Disruptions such as severe weather and pandemics alter system priorities in the short and long term (Hennig *et al.* 2022, Stern *et al.* 2021). The model utilizes scenario analysis to determine how priority rankings change due to disruption (Hassler *et al.* 2020). It is difficult to balance competing metrics, objectives, and perspectives when designing resilient systems (Hollenback *et al.* 2020). The decision model enables organizations to evaluate conflicting criteria to improve selection across the set of resilience measures (Almoghathawi *et al.* 2017, Andrews *et al.* 2020 (2)). Table 2-7 describes the literature regarding risk analysis in sociotechnical systems, logistics systems, supply chains, and similar domains.

Table 2-7. Summary of literature on risk and scenario analysis, the disruption of system orders, logistics, and supply chains

Aspect of risk or scenario analysis	References
Analysis of disruption of priority orders due to disruptive scenarios	Loose *et al.* 2023(a) Lambert, Loose *et al.* 2022 Loose *et al.* 2022(c) Thorisson and Lambert 2021

Preference modeling under uncertainty	Loose *et al*. 2022(a) Collier *et al.* 2021(c) Quenum *et al.* 2021 Bouchaut and Asveld 2020 Hassler *et al.* 2020 Haimes *et al*. 2002
Preference modeling for maritime ports	Almutairi *et al.* 2022 Gacek *et al.* 2021
Preferences, security, risk, and trust for supply chains	Eddy, Loose *et al.* 2022 Loose *et al.* 2022(b) Collier *et al.* 2021(e) Collier *et al.* 2021(f) Jensen 2021 Pennetti *et al.* 2020(c)
Systems analysis with evolving preferences	Chmura 2021 CPDC 2021 VEC 2021 Collier and Lambert 2020(c) Klasa *et al.* 2020 Keisler *et al.* 2020 Pennetti *et al.* 2020 (1)

Risk in transportation	Alsultan *et al.* 2020 Pennetti *et al.* 2020(b) Pennetti *et al.* 2020(d) Levenson 2004
Trust, security, and understanding of complex systems	Moghadasi *et al.* 2022 VanYe *et al.* 2021 Collier and Lambert 2020(a)
Analysis of stakeholder disposition changes under uncertainty	Alsultan *et al.* 2021 Bonato *et al.* 2021(b) Donnan *et al.* 2020
Resilience of ports to disruptions	Andrews *et al.* 2020(a) Thorisson *et al.* 2019(b)
Project planning and management	Eddy, Loose *et al.* 2023 Collier and Lambert 2021 Crater PDC 2021 VDEP 2021
Evolution of risk analysis, risk metrics, and resilience	Wheeler *et al.* 2022 Pennetti *et al.* 2021 Hollenback *et al.* 2020 Linkov *et al.* 2020 USBLS 2020 Hutchins 2018 Conrow 2007

Chapter 3. Methods

3.1 Overview

This chapter describes the cyber-physical systems requirements methodology (CPSRM) and a brief overview of the tools used for executing each stage. The CPSRM is a framework for the development of requirements for resilience in cyber-physical systems. Section 3.2 describes the CPSRM including the relevant context for the methods when compared to other cyber-physical systems analysis frameworks. Section 3.3 describes the four teams of the CPSRM and their roles, skills, and responsibilities. Section 3.4 describes the six sages of the CPSRM, including the relevant inputs and outputs of each stage. Section 3.5 describes the tools and methods used to execute each stage of the CPSRM.

3.2 Cyber-physical systems requirements methodology

This dissertation introduces the cyber-physical systems requirements methodology as a technique for describing, developing, and analyzing resilient cyber physical systems. The CPSRM is an extension of the cyber resilience requirements methodology (Carter *et al.* 2019). The CPSRM is a six-stage process involving four teams: T1 - purpose, T2 - function, T3 - blue, and T4 - red. Figure 3-1 describes the CPSRM stages, teams, and workflow.

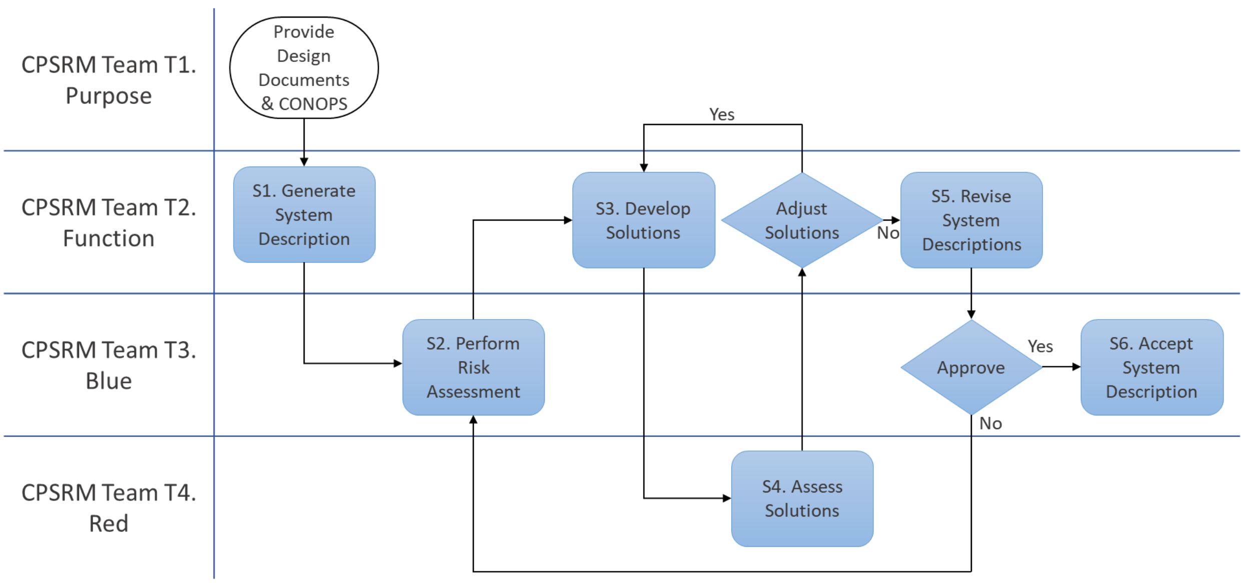

Figure 3-1. The workflow and swimlane diagram for the Cyber-Physical Systems Requirements Methodology. The CPSRM is executed across six stages and four teams.

The CPSRM and related terms and teams are derived from cybersecurity research (Chowdhury and Gkioulos 2023). Cybersecurity analysis and design methods are critical for minimizing the risks of cyber-threats to digital systems. Cyber systems span multiple user groups, objectives, and design paradigms – a cohesive design philosophy enables system owners to tailor design elements to the objectives, needs, and concerns of the user base. The principles developed for cybersecurity are expanded to apply for any cyber-physical system using the CPSRM.

Latest research focuses on the analysis and design of cyber-physical systems for resilience (Cassottana *et al.* 2023). Effective CPS design for resilience has three primary parts; 1) a description of the CPS, 2) an analysis of disruptive scenarios, and 3) the development of resilience techniques. Figure 3-2 describes the framework for analysis and design of CPS.

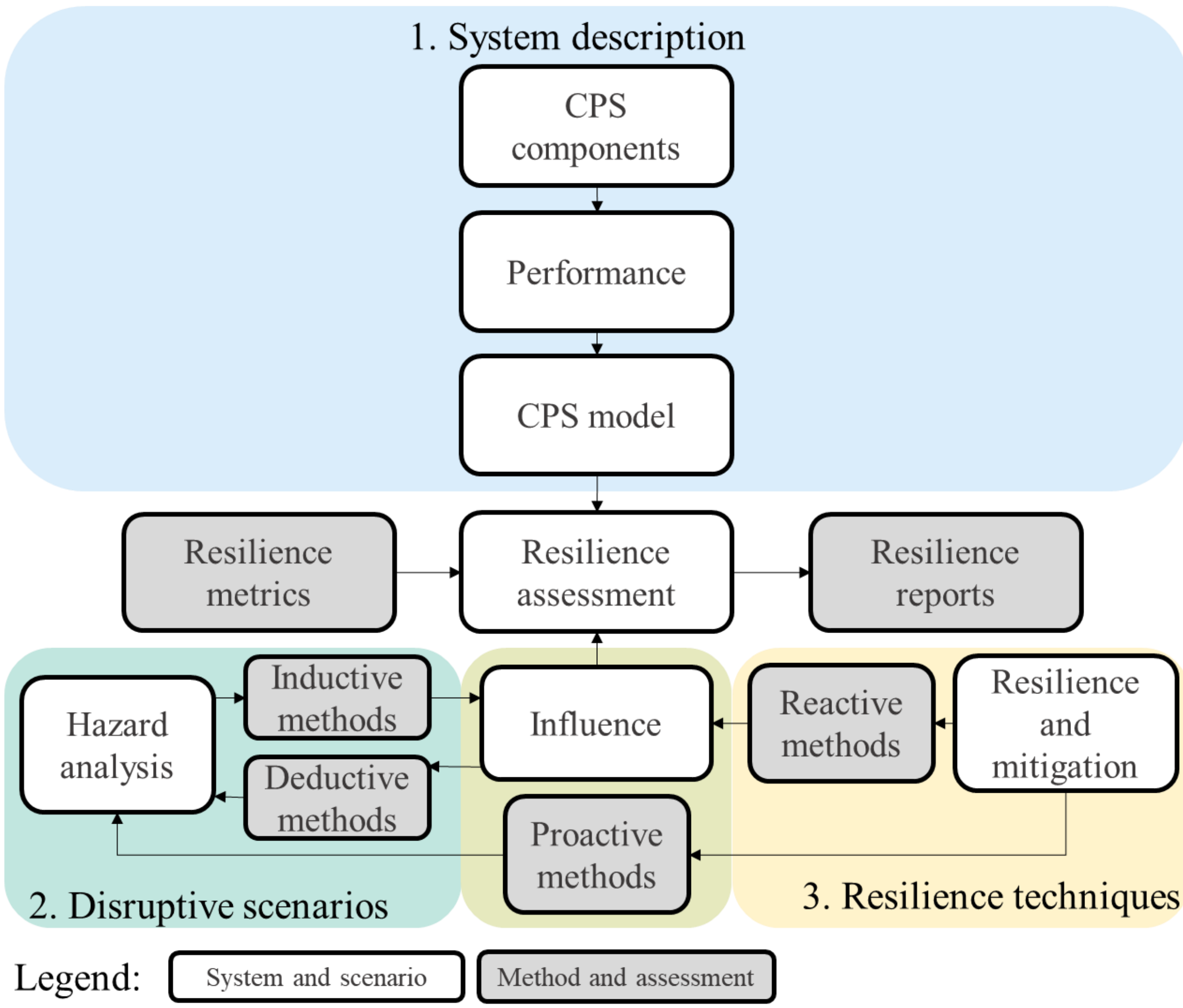

Figure 3-2. Representation of the critical phases of a cyber-physical systems design and analysis technique. There are three key parts – a system description, analysis of disruptive scenarios, and an assessment of resilience techniques. Adapted from (Cassottana *et al.* 2023).

Part one – *system description* has three elements. First, the individual CPS components are assessed, including the cyber and physical layers. This includes the collection of data and knowledge about system missions, structures, and interdependencies. Once components are collected, analysts identify measures of performance – quantitative values used to assess how the system operates under normal and disrupted conditions. The system components and measures of performance are converted into a model – this could be a simulation, statistical model, or represented in a modeling language.

There are three elements of part two – *disruptive scenarios*. The objective of part two is to perform a hazard analysis in which information about possible hazards that influence the measure of performance is collected. Hazards can be developed via *inductive methods* in which analysts predict influence of known hazards on performance. Hazards can also be developed via *deductive methods* in which disruptive scenarios are traced backwards – from the impact of the disruption to the triggering event.

Part three – *resilience techniques* is the assessment and development of strategies that can mitigate or prevent damages caused by disruptive scenarios. A resilience technique is a particular technology, process improvement, program, or structure that improves system resilience. In this part, information about existing resilience and mitigation techniques is collected. New resilience techniques may be researched, tested, and implemented in this part. Techniques can be reactive or proactive. Reactive methods are implemented to return to normal operations after disruption has occurred. Proactive methods are used to prevent or reduce the influence of disruptive scenarios on operations.

The CPSRM encompasses all three parts of the CPS resilience assessment framework, corresponding to the six phases. Preliminary work and stage *S1 – generate system description* correlates to part one – *system description*. Stage *S2 – perform risk assessment* correlated to part two – *disruptive scenarios*. Stages S3 through S6 correlate to the development of part three - *resilience techniques*. Figure 3-3 demonstrates which stages of the CPSRM correspond to which parts of the CPS resilience assessment framework. The three colors in Figure 3-3 correspond with the colors of the three parts of a CPS design and resilience technique from Figure 3-2.

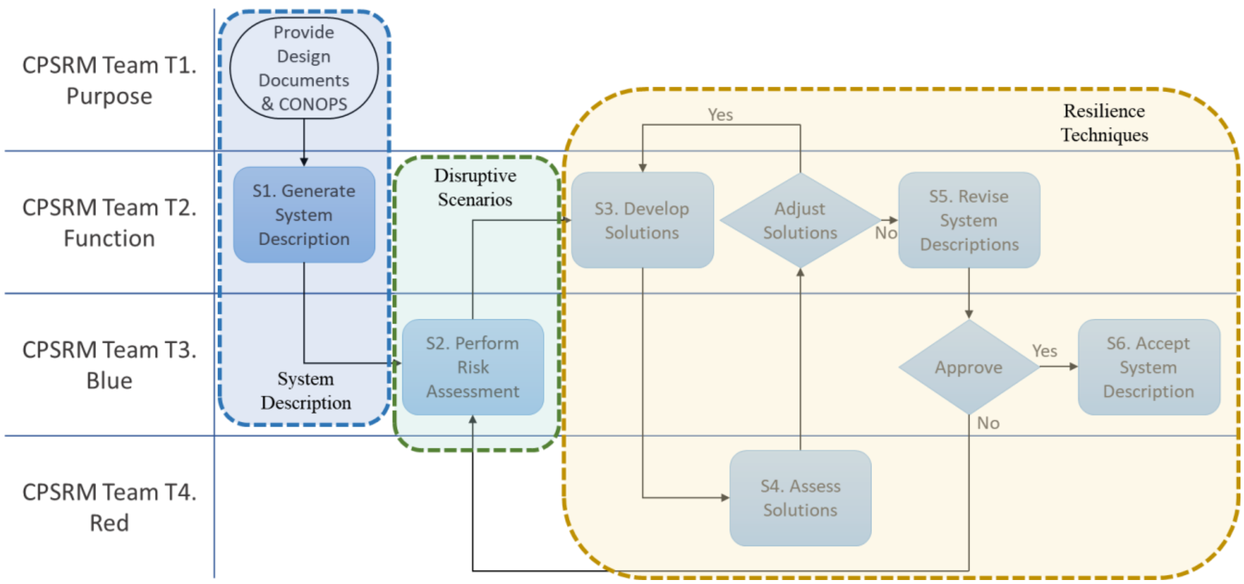

Figure 3-3. Relationship between the CPSRM and the CPS resilience assessment framework.

While the CPS resilience assessment framework provides guidelines for how to analyze and design resilient systems, the specific tools, models, and methods depend on the target system and the needs of stakeholders. Table 3-1 describes the objects of analysis and tools, methods, or models used for the demonstration of the CPSRM in this dissertation. Other tools may include graph-theory models, finite-state machines, event trees, attack graphs, Markov processes, and game-theoretical models.

Table 3-1. Overview of the tools, models, and methods of the CPSRM and the relationship to the phases of CPS design

Resilient design for CPS	Object of analysis	Tool/Model/Method
System description	Control components Cyber components	Simulation-based models
Disruptive scenarios	Physical disruptions Cyber disruptions	Contingency analysis
Resilience techniques	Hardening components Automation	Analytical models Control theory-based models Reinforcement learning

In this dissertation, the CPSRM is used to analyze a critical element of the container handling process in the Port of Virginia. A simulation-based model is used to emulate the movement of physical components and the system controller. A contingency analysis is used to identify disruptive scenarios and their impacts on operations. Reinforcement learning models, control theory models, and analytical models are used to implement resilience techniques.

3.3 Review of the teams of the CPSRM

There are four teams that interact throughout the CPSRM. These are teams with varied expertise regarding the target system, disruption, and modeling capabilities. Utilizing cross functional teams as part of requirements elicitation is a best practice in systems design and analysis (SeBOK 2023). Varied experience across teams allows for greater depth and breadth of analysis and more robust results.

The first team T1 consists of stakeholders and experts on the target system and define the system purpose. T1 may be users of the system, maintenance personnel, management, or customers. T1 members have a vested interest in the continued performance of the system. T1 may be questioned or provide insight into how the system is used, identify objectives of the system, or describe limitations of the system. Experts are personnel who have in-depth and first-hand knowledge of the system including operational procedures. Members of T1 may be a part of the remaining three teams of the CPSRM. T1 provides the information required to develop a system description. This includes the initial design documents of the system, a record of changes to the system, and other relevant information. They may provide a Concept of Operations (CONOPS) – a plain language description of the missions, operations, and mechanics of the system. A CONOPS defines the scope of operations for users under normal operating conditions. A CONOPS may also describe design constraints, providing context for why various features were implemented or excluded. A CONOPS provides the high-level description of the functionality of the system including inputs, outputs, processes, and feedback (Laarni *et al.* 2022).

The second team is T2 and consists of systems design experts responsible for defining system functions. T2 is comprised of analysts with cross-functional skillsets including risk and resilience analysts, programmers, modelers, and other domains. The primary responsibility of the T2 is to design and implement a model of the target system based on the documentation provided by T1. T2 uses the model to design and test resilience techniques, utilizing feedback from the red and blue teams. T2 team generates a system description in stage one of the CPSRM – a set of high-level requirements for the model that can describe the inputs, outputs, architectures, and internal dynamics of the system. T2 team uses this model to implement resilience techniques, generating new requirements for the system in stages three and five of the CPSRM.

The third team is the blue team. The blue team is comprised of operations-oriented stakeholders who create the finalized set of system requirements. Members of the blue team are typically system owners or the primary users. The blue team leverages the expertise of its members who have operated the target system in both typical and disrupted conditions, making them the ideal team for identifying and collecting disruptive scenarios and hazards to the system. Disruptive scenarios are possible future states of the system that influence operational performance and disrupt system priorities. The blue team performs a risk assessment in stage two of the CPSRM, providing information and requirements regarding disruptive scenarios to T2 as they develop the model of the system. In stage six of the CPSRM the blue team selects the final set of new resilience requirements to implement into the real-world version of the target CPS.

The fourth team is the red team. The red team is comprised of experts on disruption and resilience who can provide feedback on the efficacy of resilience techniques to T2. In traditional cybersecurity tabletop analysis, the red team represents hackers and other cyber vulnerability experts. In the CPSRM, the red team represents experts on hazard analysis and resilience. The red team assesses the performance of the resilience techniques that T2 implements, providing feedback as necessary. Table 3-2 provides an overview of the four teams in the CPSRM.

Table 3-2. Overview of the four teams of the CPSRM and associated responsibilities

Team	Description
T1 - Purpose	T1 are system owners, analysts, and designers who can provide insight to the purpose, scope, and function of the system. This team may share personnel with the three other teams of the CPSRM. The purpose of this team is to provide design documents, the Concept of Operations for the system, and other relevant information to assist the remaining teams with the development of the system description, disruptive scenarios, and resilience techniques.
T2 – Function	T2 consists of analysts with cross-functional skillsets who are responsible for the design and implementation of a systems model for testing and analysis. T2 uses information from the stakeholders and experts to identify system requirements. T2 team develops a system description, then uses risk assessment information to build a model of the system including resilience techniques. T2 coordinates with the red team to refine results and develop new resilience techniques.
T3 - Blue	The blue team consists of operations-oriented stakeholders who are experts on the target system with an understanding of disruptions to the system. The blue team performs the operational risk assessment, which includes the collection and analysis of disruptive scenarios and their impact on the system design. The blue team finalizes the system description and requirements.
T4 - Red	The red team is a group of hazard, attack, and resilience experts who identify and address specific risks. The main focus of the red team is to provide feedback to T2 during solution development, giving insight to the efficacy of resilience techniques.

3.4 Review of the stages of the CPSRM

There are six stages in the CPSRM; *S1 – generate system description, S2 – perform risk assessment, S3 – develop solutions, S4 – assess solutions, S5 – revise system descriptions*, and *S6 – accept system description*. Some stages are executed multiple times based on feedback. The goal of executing the CPSRM is to generate new requirements that improve the resilience of the target system to disruptive scenarios. The initial input to the system is a description of the high-level goals and operations of the system, and the final output is a set of new requirements for implementation with an emphasis on new resilience techniques.

In *S1 – generate system description*, T2 converts design documents and the CONOPS into high-level requirements for the system model. The objective is to generate a model of the system that can be analyzed or manipulated to understand the impacts of disruptive scenarios. The system description includes the inputs, outputs, and goals of the system model. The system description defines the measures of performance used to assess the effectiveness of the system. This may include existing resilience techniques to be modeled in later phases. T2 may begin to consider new resilience methods at this stage.

In *S2 – perform risk assessment*, the blue team identifies disruptive scenarios that may influence the performance of the system based on the system description. The primary tool of this stage is a hazard analysis – an inductive method in which the blue team iterates over a list of known disruptions, ranking each based on its ability to disrupt operations. This can include the probability of a disruptive scenario occurring. Existing resilience techniques can raise or lower the rank of a disruptive scenario – if the system is already well protected against flooding, the risk rank of flooding may decrease. Thus, it is necessary to also perform a gap analysis. In this case, a gap analysis is a review of existing resilience techniques and strategies that are already applied to the system. Once the list of existing techniques is set, the blue team identifies new resilience techniques that can be implemented to address disruptive scenarios. The blue team then identifies "gaps", or a set of new techniques or technologies than can be implemented to improve system resilience.

In *S3 – develop solutions*, T2 converts the system description into a model. This can be a graph theory model, dynamic system model, Petri net, or other type of system representation. In this dissertation, T2 utilizes a simulation-based model. The model is a representation of the real-world CPS, and is used to test new resilience techniques. The model should be able to integrate the effects of disruptive scenarios. The model should be functional, but does not need to be complete – the model is sent to the red team in stage four for further assessment.

In *S4 – assess solutions*, the red team analyzes the model, paying particular attention to the effectiveness of the resilience techniques. They analyze the measures of performance across scenarios to determine how effective the techniques are in improving system resilience. Using their expertise on hazards, the red team provides feedback on potential changes for the model to T2, who will adjust and develop new solutions accordingly. This process can be completed as many times as necessary. In this dissertation, the red team provides feedback on the parameters of the reinforcement learning model for the container handling process.

In *S5 – revise system descriptions*, T2 rewrites the system description incorporating the new resilience techniques developed in stages three and four. There may be one or more new techniques implemented at one time. The revised description is given to the blue team who approves the new requirements, updating the system description.

In *S6 – accept system description*, the blue team finalizes the updated system description and prepares to implement the new resilience techniques. If the blue team rejects the revised system description the CPSRM restarts in *S2 – perform risk assessment*. This way, the blue team can perform the gap analysis again using new context provided in the updated system description. It is not necessary that the blue team accept all new resilience techniques or requirements. The blue team can perform a metanalysis of the new requirements, sorting them by priority and assessing how priorities change under disruptive scenarios as outlined in the hazard analysis. Through this process, the blue team gains clarity on which disruptive scenarios are most relevant and which requirements are the highest priority. Table 3-3 provides an overview of the six stages of the CPSRM. Table 3-4 provides an overview of the inputs and outputs required for each stage of the CPSRM

Table 3-3. Overview of the six stages of the CPSRM and descriptions

Stage	Description
S1 – Generate System Description	In stage one, T2 generates the system description – a high-level overview of system goals, components, inputs, outputs, and strategies. These are the components of the system that will be modeled in subsequent steps of the CPSRM.
S2 – Perform Risk Assessment	In stage two, the blue team performs a risk assessment to identify scenarios that may disrupt the performance of the system. This includes a gap analysis to understand which capabilities the system utilizes to reduce the impact of disruption and what new capabilities may improve system resilience.
S3 – Develop Solutions	In stage three, T2 creates a model of the target system using the system description and input on disruptive scenarios from the blue team. This model is used to test resilience techniques and generate new resilience requirements
S4 – Assess Solutions	In stage four, the red team analyzes the models of T2 to assess the efficacy of the resilience techniques. They provide feedback T2 so they can adjust the model accordingly
S5 – Revise System Descriptions	In stage five, T2 updates the model and resilience solutions to address the feedback from the red team. Stages three and four are repeated as needed.
S6 – Accept System Description	In stage six, the blue team accepts or rejects the new requirements. The blue team determines which new requirements will be implemented in the system, especially with respect to resilience requirements addressing disruptive scenarios. The blue team is interested in understanding the ordered priority of the requirements and how different disruptive scenarios influence this priority.

Table 3-4. Descriptions of the inputs and outputs of each stage of the CPSRM

Stage	Inputs	Outputs
S1 – Generate System Description	Design documents, CONOPS, expert advice	System objectives and specification, preliminary requirements
S2 – Perform Risk Assessment	System objectives, specification, and requirements; Information on disruptive scenarios	Assessment of existing resilience capabilities, gaps in resilient design, resilience requirements
S3 – Develop Solutions	System specification, resilience requirements	Model of target system, including resilient design components
S4 – Assess Solutions	Model of system, including resilient design components	Feedback regarding efficacy of resilient components, including how well they address disruptive scenarios
S5 – Revise System Descriptions	Feedback from red team	Updated system description
S6 – Accept System Description	Updated system description	Finalized list of new resilience requirements for the target system

3.5 Tools and methods used for the execution of the CPSRM

Each stage of the CPSRM utilizes different sets of analysis tools and methods. This section focuses on the tools used in this dissertation to execute a demonstration of the CPSRM on a maritime port and surrounding region, specifically the container handling process.

S1 – generate system description utilizes the system specification process – also called the system requirement specification (Suhail *et al.* 2022). A system specification is a document that outlines the expected behavior and outcomes of the system. A system specification includes the purpose of the system (what does the system do?) and the need for the system (why should the system be built?). The system specification may include preliminary functional requirements – a set of the capabilities of the system. These may be things that the system *will* and *will not* do, setting boundaries on functionality. A system specification also includes a description of the existing system, including what currently exists, what the new requirements may be, and who the users will be. Other information in the system specification includes the constraints of the system (timelines, technical software limitations, legal obligations) and assumptions regarding dependencies that may impact development (Verasseti 2021).

S2 – perform risk assessment utilizes two methods – a hazard analysis and gap analysis. A hazard analysis is the process by which experts enumerate as many of the known and relevant hazards as they can, and determining which hazards pose the greatest risk to operations. This process also highlights the *impacts* of the disruptions – that is, multiple hazards may have the same influence on system performance. A gap analysis consists of taking inventory of existing resilience techniques and identifying a desired target state for the system. The difference between the *as-is* system and the target *to-be* system is called a *gap*. The gaps are presented as requirements for inclusion in the system model. The gaps are presented to T2 for inclusion in the simulation.

S3 – develop solutions utilizes simulation and reinforcement learning algorithms. The simulation is implemented in Python and provides a digital representation of the environment, actions, and internal dynamics of the system. It also incorporates randomness, emulating a real-world environment. This randomness can be manipulated to resemble the impacts of a disruptive scenario. Reinforcement learning is used to train control algorithms, specifically the implementation of the MuZero algorithm. The algorithm learns how to respond to various system states, improving decision making. The algorithm also learns the underlying dynamics of the simulation, allowing it to more effectively respond to disruption.

S4 – assess solutions and *S5 – revise system descriptions* both iterate on the algorithm, selecting hyperparameters that best match the system. This may include changes to the simulation to more closely resemble the real-world system. This process includes monitoring the measure of performance as well as monitoring the performance of the RL algorithm.

S6 – accept system description utilizes a mathematical analysis framework to prioritize resilience requirements and understand how this priority changes under disruption. This produces two main artifacts. First, the framework produces an analysis of priority orders, indicating how individual priorities change due to disruption. A new requirement may be high priority in the baseline, but falls in importance due to disruption. Second, the framework quantifies how disruptive scenarios are, allowing the blue team to see which disruptive scenarios influence priorities most and least. Table 3-5 describes the tools and methods associated with each stage of the CPSRM, including the chapters of this dissertation that discuss these tools.

Table 3-5. Relationship between CPSRM stages, chapters, and tools or methods

Stage	Chapter	Tool(s) or Method(s)
S1 – Generate System Description	Chapter 4	System Specification
S2 – Perform Risk Assessment	Chapter 4	Hazard Assessment and Gap Analysis
S3 – Develop Solutions	Chapter 5	System Simulation and Reinforcement Learning Algorithm
S4 – Assess Solutions	Chapter 6	Train Algorithm, Assess Results
S5 – Revise System Descriptions	Chapter 6	Train Algorithm, Assess Results
S6 – Accept System Description	Chapter 7	Mathematical analysis of disruption of priorities

Chapter 4: System specification and gap analysis

4.1 Overview

This chapter describes the development of a system specification of the Port of Virginia and surrounding region. A key element of this is the development of a hazard and gap analysis to provide context for the risks to the port. Section 4.2 provides an overview and description of the system used for the demonstration of the CPSRM, including the port, its missions, and the surrounding region. Section 4.3 provides an overview of the system specification of the port, including section 4.3.1 providing details of the Port of Virginia and its macro-objectives, and section 4.3.2 providing details of the specific port processes analyzed in this demonstration. Section 4.4 provides details on the hazard analysis, the first phase of the gap analysis, including Section 4.4.1 providing an overview of the hazard analysis process, and Section 4.4.2 providing a description of the hazards. Section 4.5 describes the gap analysis, the process through which new resilience techniques are assessed for viability in the updated system specification.

This chapter represents stages one and two of the CPSRM. Prior to the execution of the CPSRM, T1 delivers design documents and other requirements to T2. In stage *S1 – Generate System Description*, T2 utilizes the design documents to develop a high-level system description. In stage *S2 – Perform Risk Assessment,* the blue team performs a risk assessment by developing a hazard analysis and gap analysis. Figure 4-1 describes the relationship between the content of this chapter and the CPSRM.

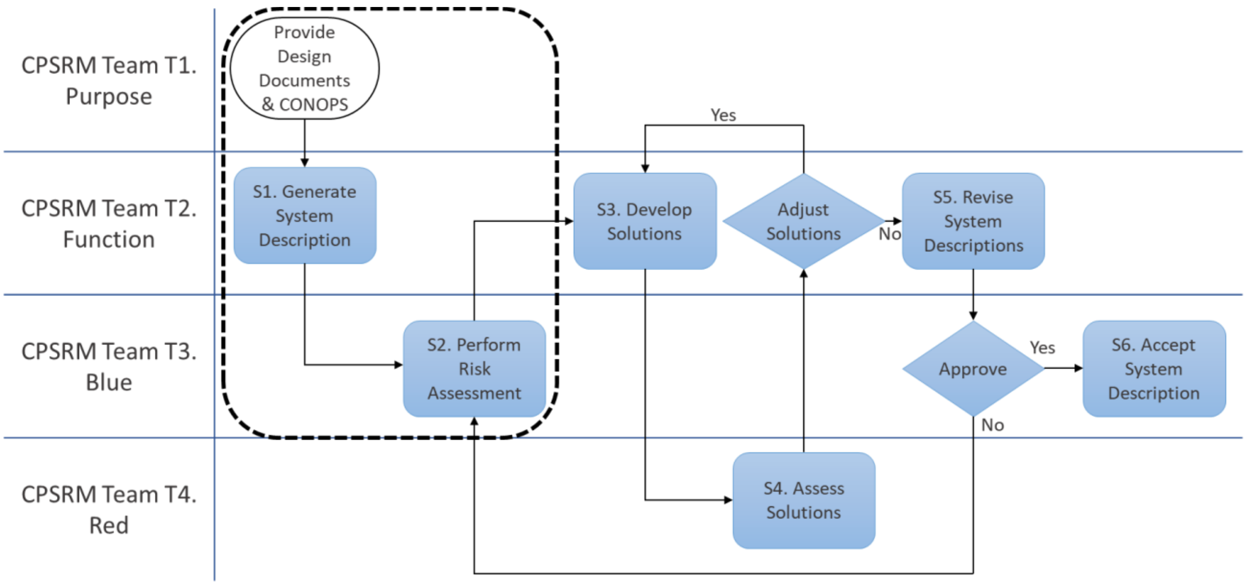

Figure 4-1. Relationship between CPSRM stages one and two, and Chapter 4 – System specification and gap analysis. Chapter 4 describes stages one and two of the CPSRM, in which the T2 develops a system description in the form of a system specification, and the blue team performs a risk assessment in the form of a hazard and gap analysis

4.2 Description of system used for demonstrations

This section provides an overview of the system of analysis in this dissertation. The primary focus is on the Port of Virginia. As the port is beholden to local, state, and federal regulations, these aspects are analyzed as well. This section provides an overview of the state of Virginia, the Hampton Roads region, and Port of Virginia facilities.

The Commonwealth of Virginia is on the Mid-Atlantic coast of the United States. The western portion of the state overlaps with the Blue Ridge and Cumberland Mountains. The central portion of the state, the Piedmont region, is a plateau bounded by the coastal plains to the east and the Blue Ridge Mountains to the west. The eastern portion of the state is bordered by the Chesapeake Bay and Atlantic Ocean. The Port of Virginia falls in the Outer Coastal Plain, and is strategically located at the intersection of the Chesapeake Bay and Atlantic Ocean (DCR 2021).

Virginia has a population of over 8.3 million as of 2022. The three most populous regions in Virginia are Northern Virginia (The Washington, D.C. metropolitan area), Richmond, and Hampton Roads (the location of the Port of Virginia) – over 70% of the population of the state lives in one of these regions. These regions are critical to the economy of the state. The remainder of the population is largely rural, with several small urban centers throughout. The Port of Virginia is strategically and practically important to the state – goods arriving through the port are distributed to the dense population areas via rail and truck. As the seat of the federal government, consistent supplies of goods to Northern Virginia are critical. Further, as the Port of Virginia is located in one of the most populous areas in the state, the port has a responsibility to reduce emissions and develop sustainable operations to ensure the continued health and wellbeing of nearby citizens. Figure 4-2 describes the population density of Virginia.

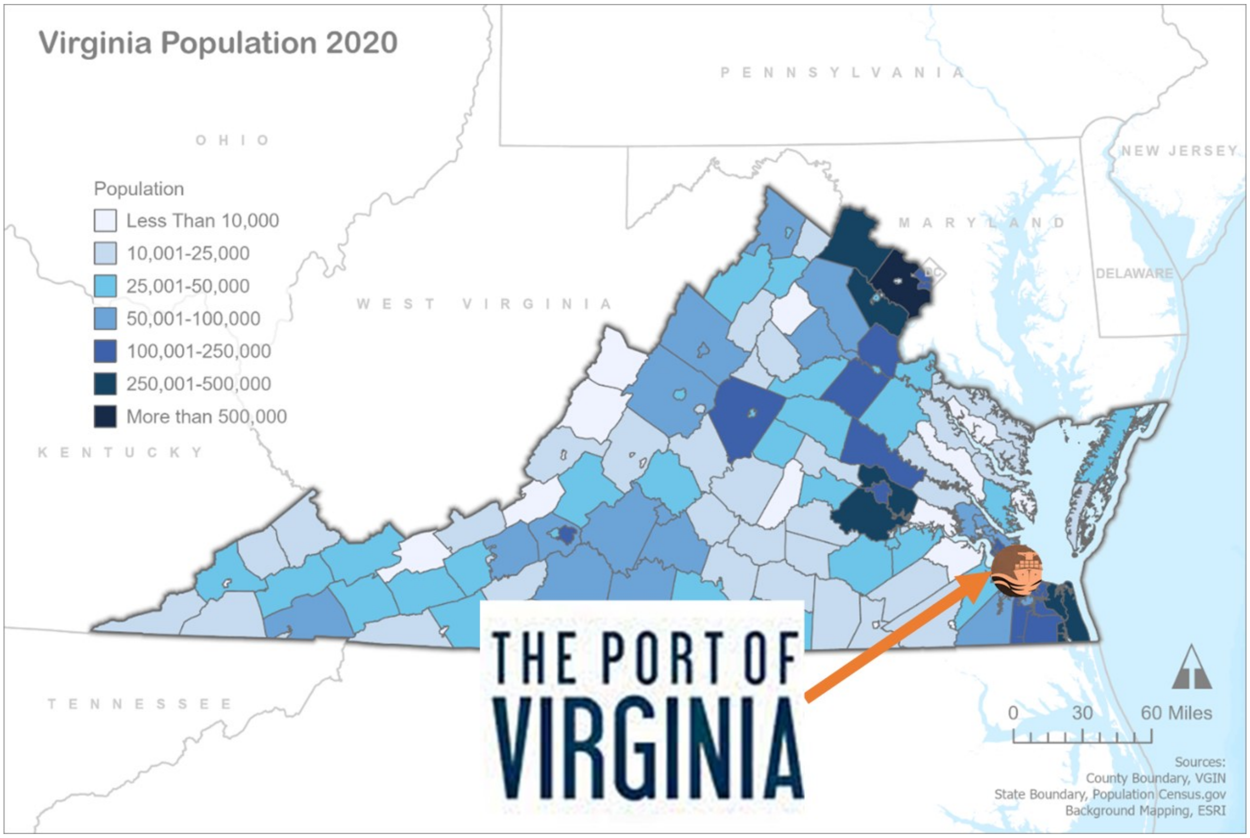

Figure 4-2. A population density map of the Commonwealth of Virginia in 2020. The map highlights the location of the main Port of Virginia terminals in the southeast corner of the state.

The state is divided into dozens to hundreds of administrative regions, including cities, counties, and federal land. Hazard analysis is conducted at the planning district commission (PDC) level. PDCs are associations of local city and county governments that coordinate projects and resources to benefit the region. There are 21 PDCs in Virginia. The four largest terminals of the Port of Virginia are located in the Hampton Roads PDC. While some facilities such as the Richmond Marine Terminal fall outside of the Hampton Roads PDC, this hazard analysis focuses on the four primary port terminals. Figure 4-3 shows the borders of the 21 PDCs of Virginia and highlights the location of the Port of Virginia within the state.

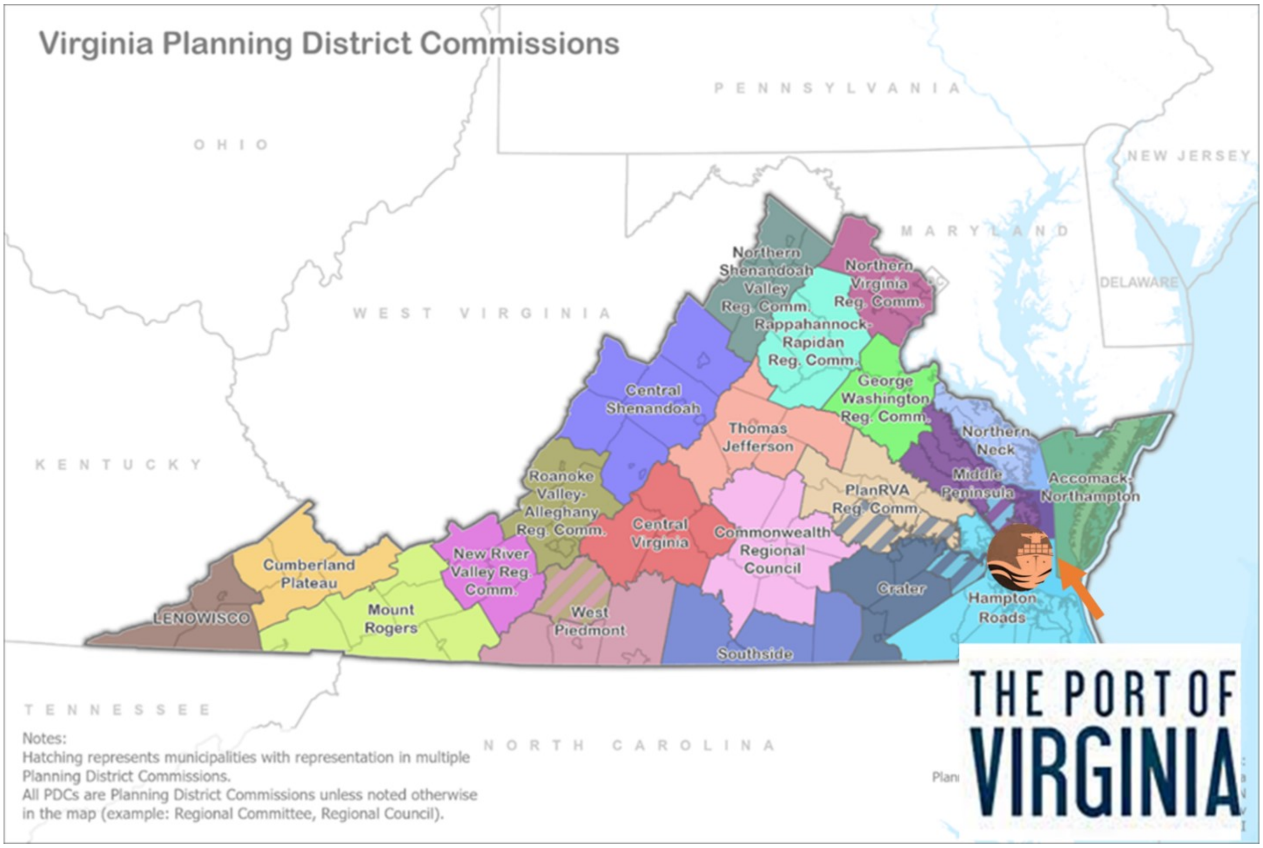

Figure 4-3. A map of the 21 planning district commission of the Commonwealth of Virginia. The main marine terminals of the Port of Virginia are located in the Hampton Roads PDC. Adapted from (VDEM and Loose 2023).

The Port of Virginia is a group of facilities and marine terminals in Hampton Roads managed by the Virginia Port Authority – an autonomous agency within the government of the Commonwealth of Virginia. As an agency of the state government, the Virginia Port Authority reports to the Virginia Secretary of Transportation. The Virginia Port Authority provides input and guidance to the hazard analysis for the state of Virginia, with special instruction for the Hampton Roads region around the main port facilities. The four primary port facilities are the Norfolk International Terminals (NIT), Virginia International Gateway (VIG), Portsmouth Marine Terminal (PMT), and the Newport News Marine Terminal (NNMT). Craney Island is a planned fifth facility located near the four existing terminals. With a new facility currently under development, the Port of Virginia is seeking new ideas, techniques, technologies, and practices to improve port performance and increase sustainability (POV 2023). Table 4-1 describes the four main Port of Virginia facilities as well as two auxiliary inland facilities run by the Port Authority and the Craney Island facility. Figure 4-4 provides the specific locations of the four primary facilities in Hampton Roads.

The Port of Virginia is one of the four largest ports on the Atlantic coast of the United States, and third in total containers handled. The port is strategically located between the Port of New York and New Jersey to the north, and the Georgia Ports to the south (SHIPA 2023). The mission of the Port of Virginia is to drive business, innovation, and sustainability in the Commonwealth of Virginia. These missions are reflected in specific objectives such as the reduction of truck turn times, net-zero emissions by 2040, and investments in local infrastructure (Virginia Port Authority 2021). The scope of port operations spans the four primary terminals, each with vessel, rail, truck, crane, and yard operations. Stakeholders from the Port of Virginia provided system specifications for each terminal, but this dissertation will focus on a single process – the control of rail mounted gantry cranes for the organization of container stacking blocks to improve the resilience of the container handling process.

Table 4-1. Overview of the facilities managed by the Port of Virginia

Port Facility	Description
Norfolk International Terminals (NIT)	The primary marine terminal in the Virginia Port Authority's control. Has the capacity to call the newest class of Ultra Large Container Vessels.
Virginia International Gateway (VIG)	A privately owned container terminal, and one of the Port of Virginia's first semi-automated terminals.
Portsmouth Marine Terminal (PMT)	Able to handle containers, break-bulk, and roll-on/roll-off cargo. However, PMT is currently (2023) closed to container traffic.
Newport News Marine Terminal (NNMT)	A multi-use terminal, NNMT boasts direct, on-dock rail service and the capability to house specialized cargo.
Virginia Inland Port (VIP)	An inland dry port and container transfer facility located 60 miles west of Washington, DC.
Richmond Marine Terminal (RMT)	An inland port located on the James River that specializes in handling temperature-controlled containers, break-bulk, bulk, and neo-bulk cargo.
Craney Island	A planned expansion for the Port of Virginia to provide additional cargo capacity and implement new container handling techniques.

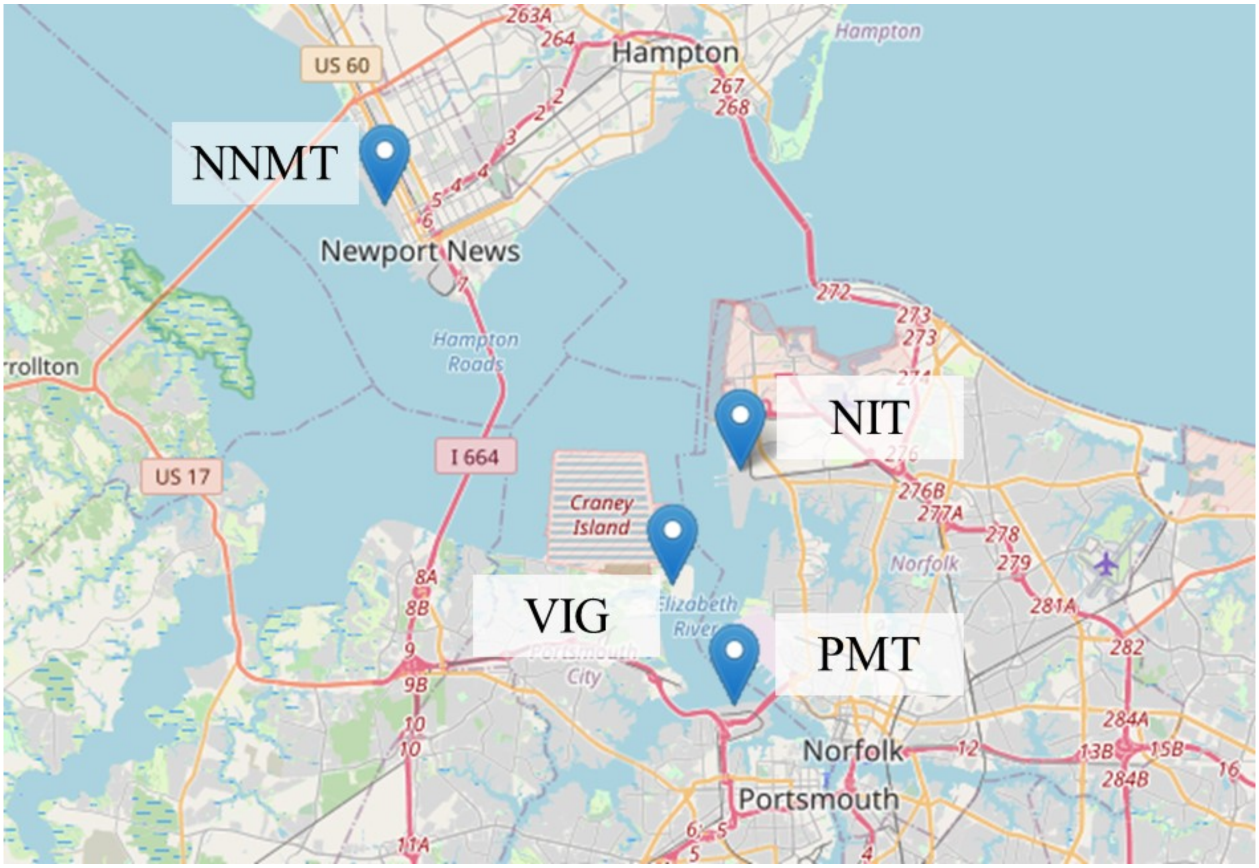

Figure 4-4. Locations of the four primary Port of Virginia facilities

4.3 System specification

This section describes the system specification (sometimes called a system requirements specification) process for the demonstration of the CPSRM on the Port of Virginia. A system specification is a structured and regimented method for collecting the requirements of a system (IEEE CS 1998). A system specification corresponds to stage one of the CPSRM, *S1 - Generate System Description*. A system specification outlines high level considerations such as system goals, missions, and objectives. It outlines the limitations on the system such as rules, regulations, and budgets. S system specification outlines the specific, actionable requirements of the system – that is “the system shall…” perform some action within a timeframe yielding a specific result.

System specifications are typically performed on subsystems or processes rather than at the organizational level. The system specification describes the operations of the system under normal operating conditions. It is an objective of the CPSRM to develop requirements that make the system resilient to disruptive scenarios, as well as improve performance. Section 4.3.1 describes the Port of Virginia at a high level, including its missions, objectives, and motivation for the development of new resilience techniques. Section 4.3.2 describes the container handling process – the specific subsystem of analysis in this dissertation – including performance metrics and desired outcomes.

4.3.1 Port of Virginia Mission and Objectives

The Port of Virginia is a maritime container and bulk cargo port with primary facilities centered on Hampton Roads, Virginia. The Port of Virginia considers the 500-mile radius surrounding port facilities to be its zone of influence. Port of Virginia operations directly impact 390,000 jobs in Virginia, representing nearly 10% of the workforce in the state. This drives nearly $21 billion in compensation and over $2 billion in taxes (POV 2023). The self-described mission of the Port of Virginia is to "[drive] business to, *and through*, the Commonwealth [of Virginia]" (PoV 2023). This is notable, as the first order objective implies an obligation first to the state of Virginia. As the Port of Virginia is operated by the Virginia Port Authority, the objectives of the state and port must align. The vision of the Port of Virginia is to leverage its diverse talent pool, technology, and industry expertise to set the standard for supply chains now and into the future. The Port of Virginia especially values innovation through technology and computing, helpfulness to the community, fortitude, accessibility to customers, mindfulness in coordination with customers and the state, and sustainability with respect to environmental impacts. These are broad missions and a guide for the creation of subsystems within the Port of Virginia at large.

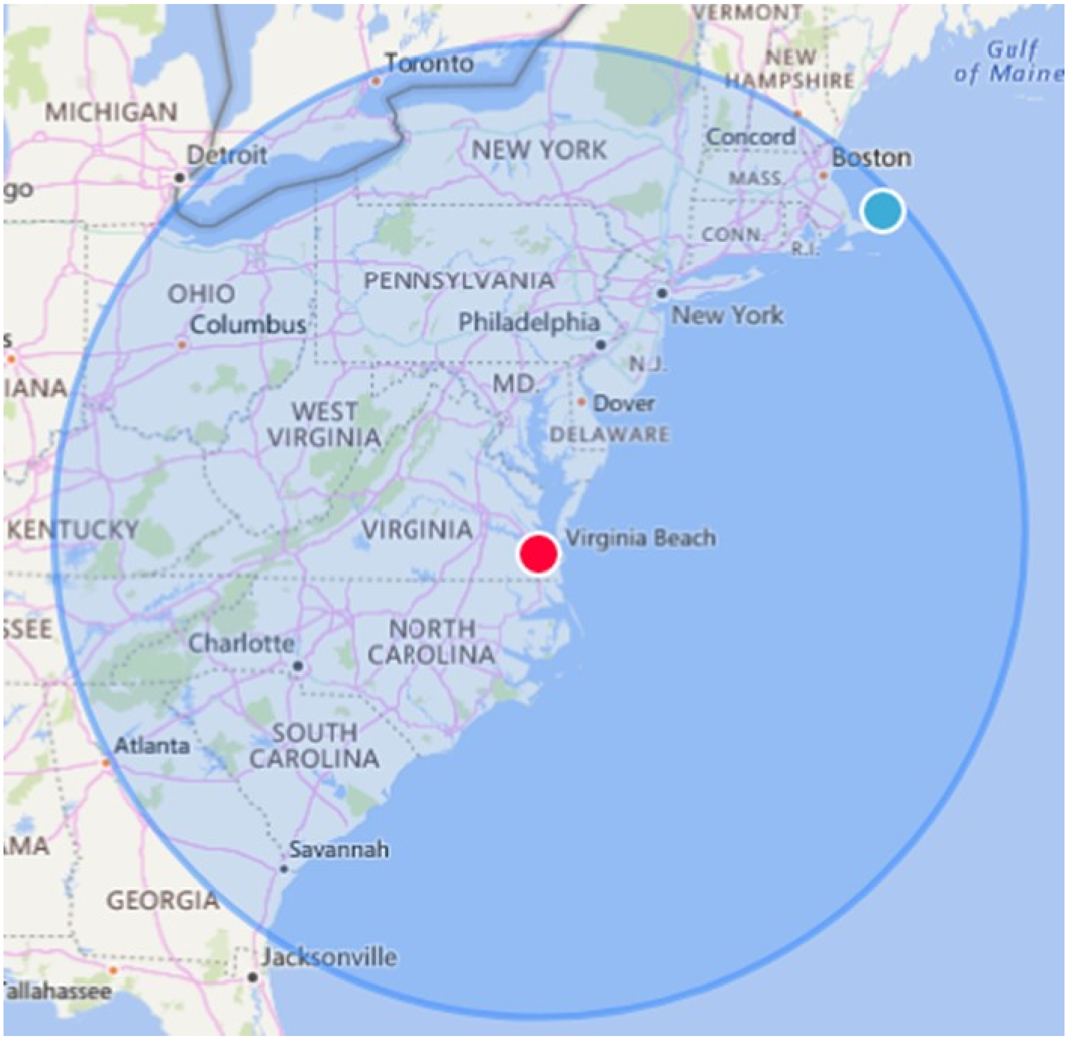

Figure 4-5. Map of the 500-mile radius zone of influence of the Port of Virginia. The port influences cities as far north as Boston, as south as Savannah, and west and Columbus. Developed using (CalcMaps 2015).

The short and medium-term goals of the port provide specific motivations for the development of enhanced technology systems. The Port of Virginia focuses on customer service with respect to process efficiency and innovation – that is, ensure customers receive prompt and reliable service and information from the Port. The Port of Virginia also has a target to become carbon neutral by 2040. Finally, the Port of Virginia desires to meet or exceed all federal, state, and local guidelines for operations. By 2065, the port plans to deliver several projects that improve operations, including the integration of new equipment and technologies to enhance port processes. These goals are fairly broad, so this dissertation will narrow the process down further to the container handling process specifically.

The container handling process is discussed in greater detail in subsequent sections, but in brief, the container handling process is the movement of cargo containers – twenty-foot equivalents (TEUs) – from vessels into the port, within the port, and out of the port. Figure 4-6 provides an image of a typical 40-ft. container, equal to two TEUs. There are many goals and objectives within the container handling process. For example, minimizing truck turn times, optimizing vessel-berth assignments, yard truck charging optimization, and minimizing container touches are a few specific goals and metrics that the Port of Virginia monitors. The demonstration of the CPSRM in this dissertation focuses on the container yard portion of the container handling process, also called the "container stacking problem". The container yard is the area within the port where TEUs are stacked and stored while awaiting pickup by truck. Storing and moving containers within this stack and during pick up by trucks is a known bottleneck – trucks often have to wait several minutes for the proper container to arrive to their vehicle. The Port of Virginia seeks methods to improve this process, reducing the average number of times a container is moved, called a "touch". This aligns with larger port mission of driving business in Virginia, reducing carbon emissions, improving customer service, meeting regulations, and minimizing container touches and truck turn times.

Figure 4-6. A 40-foot shipping container. The 40-foot shipping container (sometimes called an intermodal container) is the standardized container for intermodal freight. A 40-foot container is equal to two TEUs. Adapted from (Benutzer 2004).

4.3.2 Container yard and container handling process

This section describes the container yard and container handling process, including the measures of performance, current performance, and other important factors for consideration. This information is used to generate the requirements of the system specification. This specification is the main output of stage one of the CPSRM, *S1 - Generate System Description*. The issue at the heart of the container handling process is the container stacking block. A container staking block is a set of container stacks that are all serviced by the same rail mounted gantry (RMG) cranes.

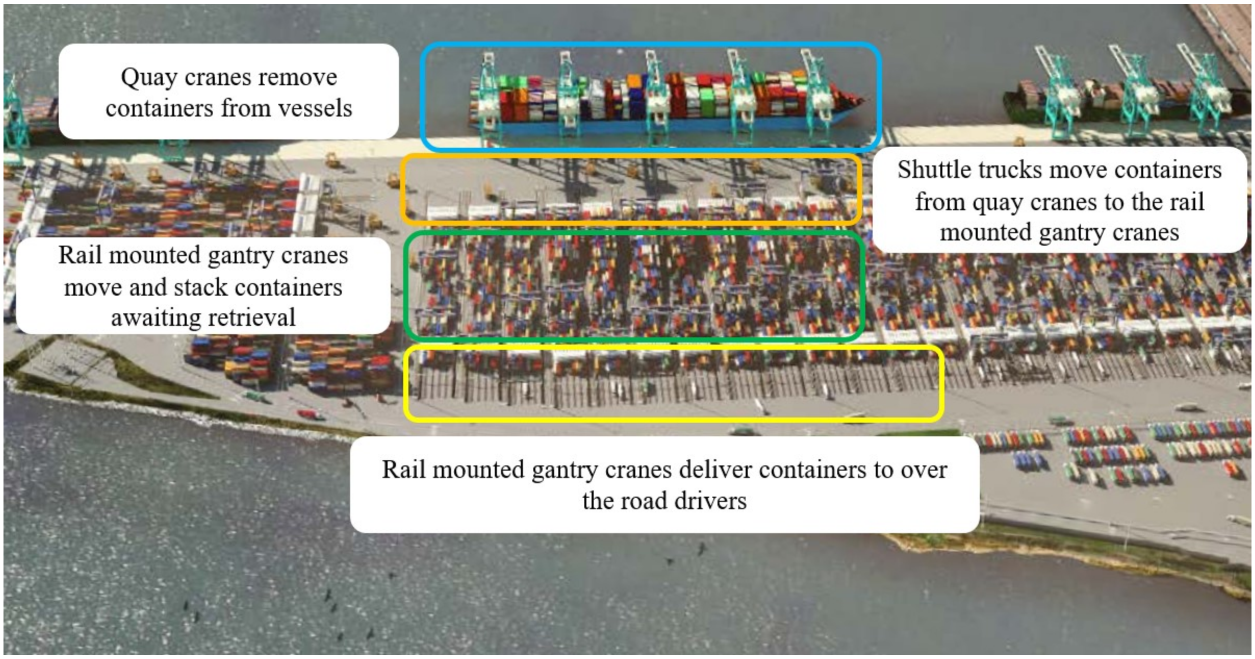

Figure 4-7. Overview of the container yard and container handling process. Adapted from (Port of Virginia 2018).

Figure 4-7 shows the major stages of container handling process processing from vessels to trucks. Quay cranes remove containers from vessels and place them in staging areas. Then, shuttle trucks move containers from staging to a container stacking block – there are multiple container stacking blocks within the container yard. RMGs place containers and manage container stack configuration. Figure 4-8 shows an RMG servicing a single container stacking block at the Port of Virginia. A container yard consists of multiple container stacking blocks. When trucks arrive, the RMGs deliver containers directly to trucks with human-in-the-loop assistance. The trucks then exit the port.

Figure 4-8. A rail mounted gantry crane servicing a container stacking block at the Port of Virginia (AAPA 2023)

The particular process of interest is the container stacking problem – the assignment of containers to locations in storage. Containers follow a probability distribution called *dwell time* in port analyses. Dwell time is the distribution of days that a given container may remain in the stacks. Current port operations utilize dwell time in decision making – a container that is likely to remain in the stacks for several more days is safe to place under containers with shorter dwell times. However, the uncertainty of when a container will leave the stacks strains effective decision making. New containers are placed atop old containers - however, this means that in many cases the newer container must be moved to a new location within the stacking block to retrieve older containers and deliver them to trucks. Ideally, a container is only moved twice – once to place the container on a stack, and a second time to deliver it to a truck. In reality, a container must be moved several times before they are removed from the stacking block. The Port of Virginia reports between 4 and 4.5 touches per container. Each move requires energy as the crane lifts and moves a container. Furthermore, a container located at the bottom of a stack when a truck arrives to retrieve it will require several minutes to reach. There are many strategies for resolving the container stacking problem. Some ports attempt to stack containers from a single customer in one stack – when trucks arrive, they can take the top container as it has the same destination as the lower containers. Others attempt to place containers in stacks in order of departure time, with sooner departure times in higher tiers (van Asperen *et al.* 2011). Both of these approaches see limited success, as it is difficult to collect data of high quality on truck arrival times, and it takes significant computing power to get information on time for delivery (Kemme 2020).

There are many considerations in the container stacking problem, often in competition with one another. Minimizing the number of moves (touches) per container would save energy and is trivial if stacks are never more than one container high. However, this would reduce the *utilization* of the stacking block – the percentage of container slots occupied by a container. This would also cause container stacks to have a much larger geographical footprint, increasing the distances RMGs travel to move containers. The distance an RMG must travel is a factor in truck turn times – in a stacking block 30 containers long, it may take a full minute to move a container from one end to the other. Moving a container at low-energy use times such as at night is less expensive than during operating hours so *reshuffles* – container movements within the stacking block – may not count against the touches per container metric.

For this demonstration, the measure of performance is touches per container. This measure aligns with the Port of Virginia goal to reduce energy usage. It is known that a stacking block with high utilization require more touches per container, but higher stacks reduce the footprint of the stacking block. Higher stack utilization and greater touches per container also increases the length of truck turn times. However, for this demonstration, it is determined that the touches per container alone is an adequate proxy for many of the other objectives of the port.

Using this information, T2 generates a high-level system description in stage one of the CPSRM. The system description centers on a single container stacking block and the associated controller. According to port documents, each container stacking block is 30 units long, eight units wide, and five units tall. Stacks are rarely completely full, though utilization levels vary with demand from 50% to 80%, though during surges the stacks can have greater than 100% utilization (stacks are allowed to be one TEU taller). Each stack utilizes two automated RMGs that can each make 40 to 50 touches per hour. Each stack services roughly 20 trucks per hour and can service up to five trucks at a time. Truck drivers can schedule pick-ups up to 48 hours from the time of their expected arrival, though nearly one-third of reservations are made same-day. For the simulation in Chapter 5, these dimensions are adjusted as part of the testing of new resilience measures for the CPSRM.

It is difficult to effectively stack containers under typical conditions. Port managers anecdotally claim that a plan made at 8:00 AM is useless by 10:00 AM. This is exacerbated when there are disruptions to operations such as severe weather. The port desires resilience techniques to reduce the impact of disruptive events with respect to managing container stacking blocks. In stage two of the CPSRM, the blue team identifies operational risks to the port to identify candidate resilience measures through a hazard analysis and gap analysis.

4.4 Hazard analysis

This section describes the hazard analysis undertaken by the blue team in stage two of the CPSRM. The hazard analysis includes an assessment of historical data to identify what future disruptive events are the most disruptive to operations. The hazard analysis presented in this section represents an analysis for the entire state of Virginia – a region within the 500-mile radius zone of influence of the Port of Virginia. Section 4.4.1 presents an overview of the hazard analysis, including important context for the hazard analysis and the methodology for selecting and ranking hazards. Section 4.4.2 describes the hazards that were analyzed.

4.4.1 Hazard analysis overview

The hazard analysis presented in this dissertation closely follows the research performed for the Commonwealth of Virginia Hazard Identification and Risk Assessment (HIRA), a key aspect of the Virginia Hazard Mitigation Plan (HMP) (VDEM, Loose 2023). The objective of this assessment is to identify the primary natural and human-caused hazards that threaten the safety of citizens in the Commonwealth of Virginia in terms of life, injury, and economic factors. The report then utilizes a gap analysis to identify existing techniques that mitigate the impacts of these hazards, as well as identify techniques that can be implemented to improve the resilience of the state. The Port of Virginia participates in the hazard analysis exercise, providing internal data on the highest risk hazards to port infrastructure as well as existing and planned mitigation actions. Though the hazard analysis provides results for the entire Commonwealth of Virginia, special attention will be paid to the Hampton Roads region, the area containing and immediately surrounding the port terminals.

Hazard analysis is an inductive method for collecting and assessing future disruptive events or conditions that impact a system. In this case, the system is the Commonwealth of Virginia with special emphasis on the Hampton Roads region. The hazard analysis consists of a regimented program of collecting existing research and consulting with experts to identify the largest threats to the state and region. These hazards are then profiled based on the history of occurrence (number of events since recordkeeping began), the vulnerability of the people in the area (population density, population vulnerability, annualized fatalities), the geographic extent of the hazard area (percent of jurisdiction in hazard area, annualized frequency), and damages in terms of property (annualized property damage in dollars, number of state facilities in hazard area) (VDEM, Loose 2023). Once the hazards are collected and profiled, they are assessed by the Advisory Committee – a council of government representatives who rank the hazards as *high*, *medium*, or *low* rank for the state as a whole. These ranks are based on point values from the hazard profiles and weights as assigned by the committee. Table 4-2 describes the results of this analysis for the Commonwealth of Virginia. The *high*-risk hazards are flooding, hurricanes, non-rotational and severe wind, winter storms, ice storms, and extreme cold.

Table 4-2. The risk rank of each hazard as described by the VDEM Advisory Committee for the Virginia 2023 Hazard Mitigation Plan. Adapted from (VDEM, Loose 2023)

Hazard Category	**Risk Rank**
Flooding	**High**
Hurricane/Tropical Storm	**High**
Non-rotational Wind	**High**
Severe Wind	**High**
Winter Storm/Weather	**High**
Ice Storm	**High**
Extreme Cold	**High**
Extreme Heat	**Medium**
Coastal/Shoreline Erosion	**Medium**
Wildfire	**Medium**
Communicable Disease	**Medium**
Tornado	**Medium**
Drought	**Medium**
Hazmat/Biological Hazard	**Medium**
Landslide	**Low**
Karst/Land Subsidence	**Low**
Earthquake	**Low**
Dam/Levee Failure	**Low**
Terrorism/Active Threat	**Low**
Space Weather	**Low**

Table 4-3 describes the results of this analysis for the Hampton Roads PDC and Port of Virginia. The *high*-risk hazards are flooding, hurricanes, and karst/land subsidence. The hazards with a risk rank of "-" and a grey background are hazards the Hampton Roads PDC either did not deem a threat to the region or did not assess. The following sections describe the identified hazards for the hazard analysis and provides insight into their impact on the state and region.

Table 4-3. The risk rank of each hazard for the Hampton Roads region of Virginia, including input from the Port of Virginia. Adapted from (VDEM and Loose 2023)

Hazard Category	**Risk Rank**
Flooding	**High**
Hurricane/Tropical Storm	**High**
Non-rotational Wind	**-**
Severe Wind	**-**
Winter Storm/Weather	**Medium**
Ice Storm	**-**
Extreme Cold	**-**
Extreme Heat	**Medium**
Coastal/Shoreline Erosion	**-**
Wildfire	**Low**
Communicable Disease	**Low**
Tornado	**Medium**
Drought	**Medium**
Hazmat/Biological Hazard	**Medium**
Landslide	**-**
Karst/Land Subsidence	**High**
Earthquake	**Low**
Dam/Levee Failure	**Low**
Terrorism/Active Threat	**-**
Space Weather	**-**

4.4.2 Hazards

Flooding is a condition in which a normally dry area is covered with water. Flooding can occur anywhere – near streams, tidal waters, rivers, or away from water sources altogether. Flooding may be caused by many factors, including thunderstorms, hurricanes, rapid icemelt, or due to the failure of levees or dams. Floods are largely unpredictable but many areas experience flooding with regularity. If these areas are inhabited or contain infrastructure, they are referred to as "repetitive loss" areas. Flooding on coasts is often caused by a storm surge – a rapid shift in sea level caused by storm winds. Flooding can cause substantial damage to large areas, rotting infrastructure, eroding the ground, and posing an immediate danger to vulnerable populations. Figure 4-9 describes the "100-year floodplain" of eastern Virginia, or the areas where there is a 1% chance of annual flooding. Floods pose a large risk to port operations, as it endangers vessels, rail, and trucks simultaneously. The Virginia HMP and Hampton Roads HMP, including the port, determined that flooding was a *high*-risk hazard.

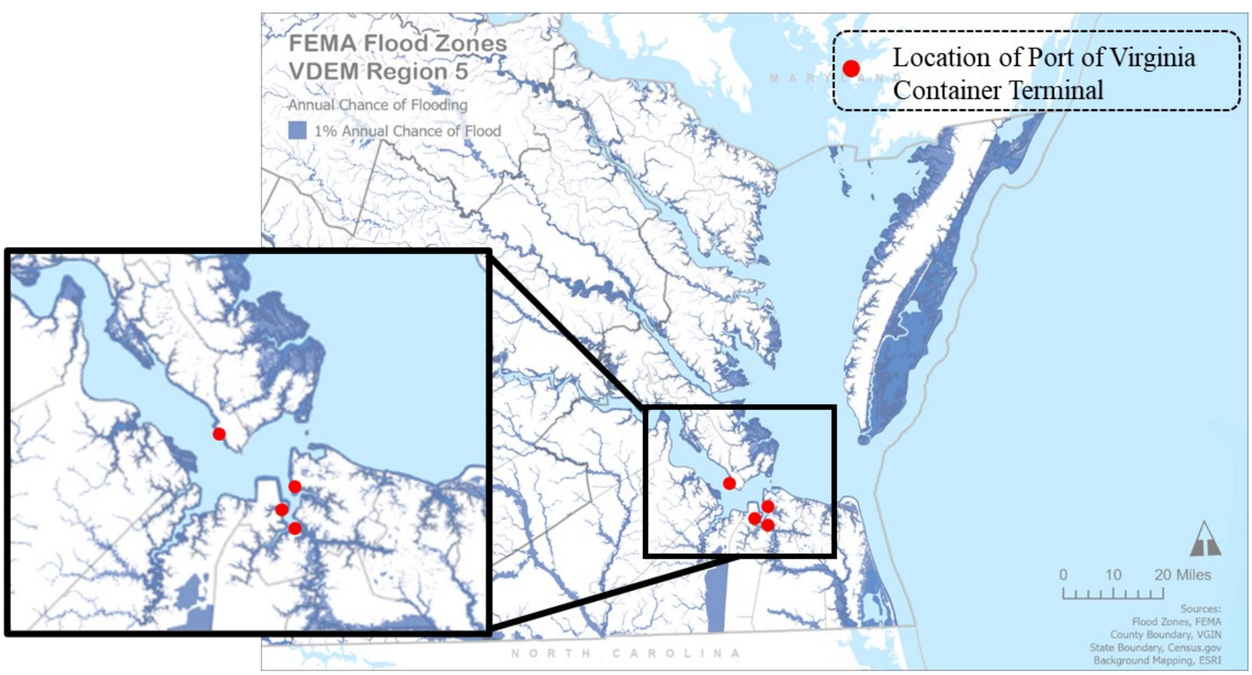

Figure 4-9. 100-year floodplain of Southeast Virginia. Adapted from (VDEM, Loose 2023)

Hurricanes and tropical storms are rotating storms with low-pressure centers, bringing high winds, heavy rains, and tornados. Rain may cause flooding and high, sustained winds can shear taller infrastructure elements. Hail and lightning are both threats during a hurricane or tropical storm event. It is possible for a hurricane to produce more than one foot of rain in a single day, leading to flash flood and mudslides. Though both coastal and inland areas are susceptible to the damage brought by hurricanes, coastal areas are especially vulnerable due the occurrence of storm surges. In Virginia, the flat topography of the coastal regions makes the area especially susceptible to storm surges. The lack of natural drainage paths leads to high accumulation of water and a slow retreat. Furthermore, the Hampton Roads region is one of the most populous and highest-risk areas for hurricane damage. The maximum of maximum envelope of high water (MEOW) is a National Hurricane Center metric for determining how far inland a storm surge may cause flooding given "perfect" storm conditions. Figure 4-10 describes the MEOW for the Hampton Roads region, including the Port of Virginia. This region is extremely susceptible to storm surges, with flooding occurring several miles inland under several conditions. Both the Virginia HMP and Hampton Roads HMP determined that the hurricane/tropical storm hazard was a *high*-risk event.

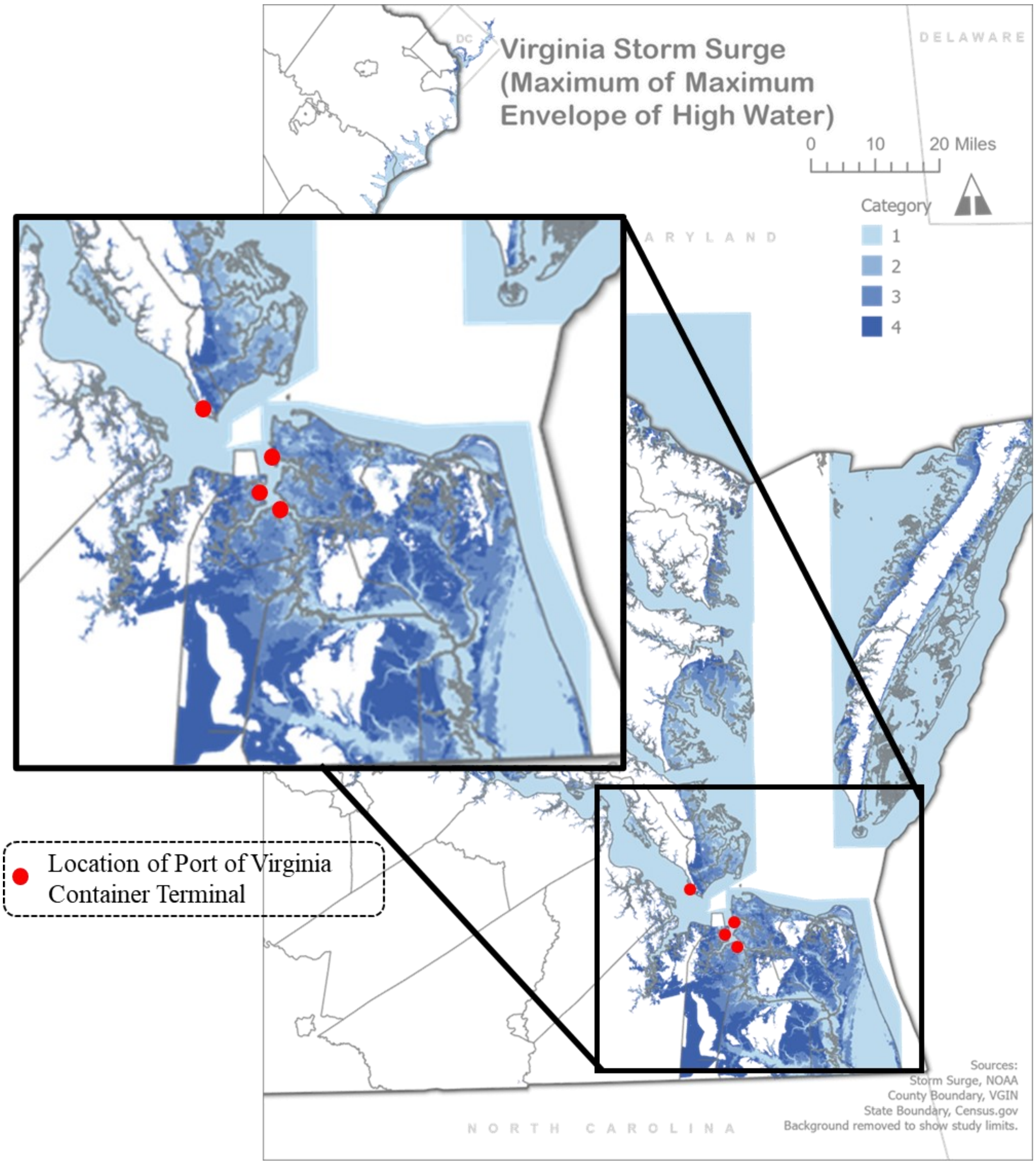

Figure 4-10. Maximum of maximum envelope of high water for category 1-4 hurricanes in the Hampton Roads region of Virginia. Adapted from (VDEM, Loose 2023)

Non-rotational wind and severe wind are winds caused by non-hurricane or tornado events such as thunderstorms, windstorms, or derechos. Derechos are wind storms caused by a long series of thunderstorms, causing damage comparable to tornados over extended periods. These wind events are uncommon in most of the Commonwealth, but have historically occurred semi-regularly in the Northern Virginia region. As this is a high-population area, the total risk of high winds is increased. Though it is less common in the Hampton Roads area, high winds are a threat to Port operations as various cranes and other port infrastructure are relatively tall and susceptible to wind shearing. The Virginia state HMP ranked the non-rotational and high wind hazards as *high* risk due to the population density, tall infrastructure, and frequency of occurrence in Northern Virginia. However, due to the low frequency of occurrence, in Hampton Roads the local HMP did not assess the high wind hazard. Figure 4-11 describes the one percent annual chance wind speeds for the Chesapeake Bay region of Virginia, including the port. The one percent annual chance of high winds in the port facilities is between 105 and 110 miles per hour.

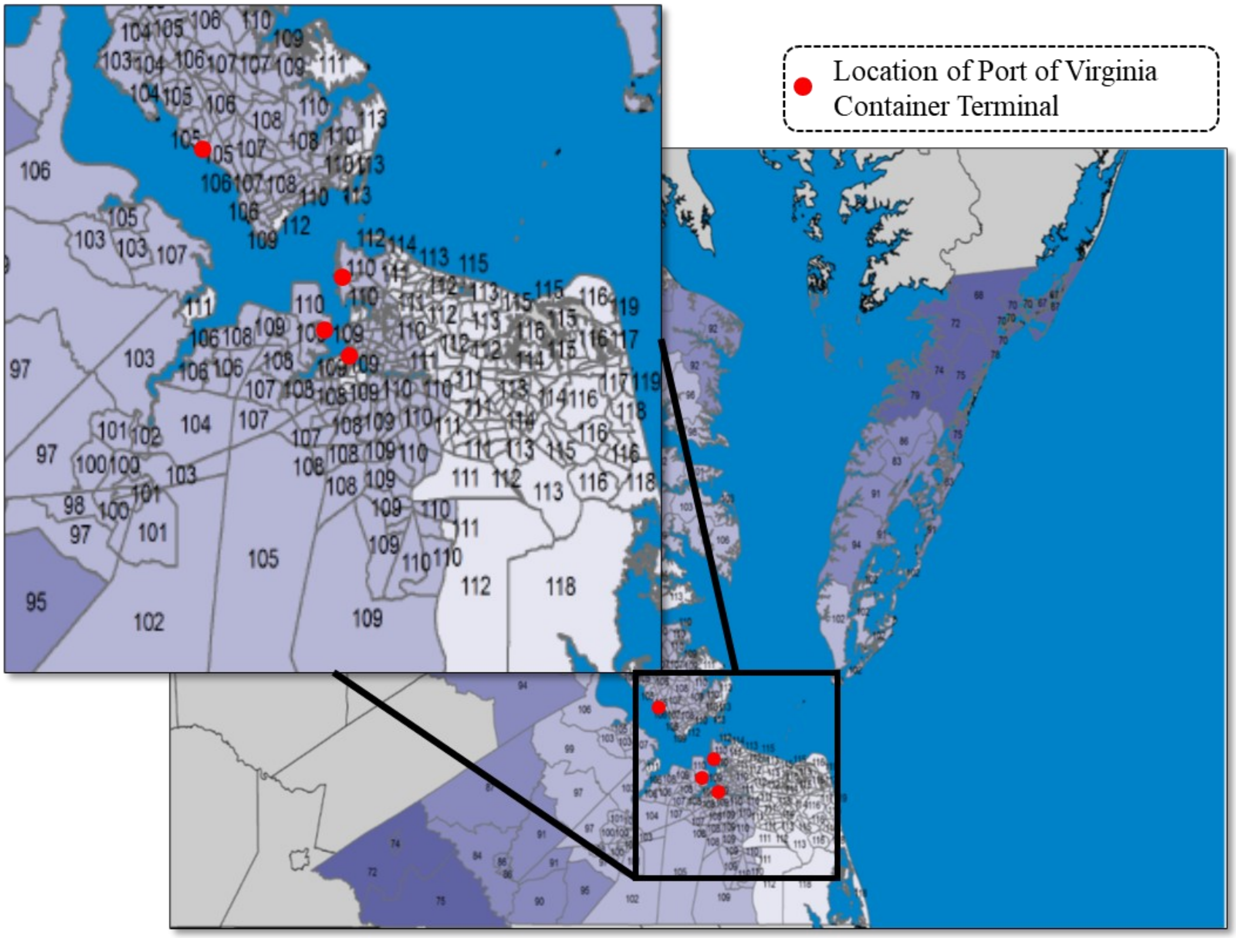

Figure 4-11. Map of the one percent annual change wind speeds in southeast Virginia. Adapted from (VDEM, Loose 2023)

Winter storms and ice storms consist of cold temperatures, snow or ice accumulation, and potentially strong winds. Winter storms can have severe impacts on roadway conditions, utility services, and the health of the populace. Winter weather events can cause days-long outages of communication, electricity, and road networks. The cold weather alone is a health risk through frostbite and freezing conditions. Accumulation of frozen precipitation has the ability to fell trees, powerlines, and other structures. The areas that most susceptible to winter storms are central and northern Virginia. Though snow accumulation is low in the Hampton Roads region, ice is common, severely disrupting road networks. Disrupted roads lead to reduced truck arrivals and longer truck turn times, severely inhibiting port operations. For these reasons, the Virginia HMP determined that winter weather and ice storms are *high* threat hazards. The Hampton Roads HMP determined that winter weather was a *medium* risk hazard. Figure 4-12 describes the winter weather risks for Virginia, with special attention to the Hampton Roads and Port of Virginia regions. The area is at low or very low risk to the population. However, winter weather is disruptive to supply chains and logistics systems, increasing the risk level for the region.

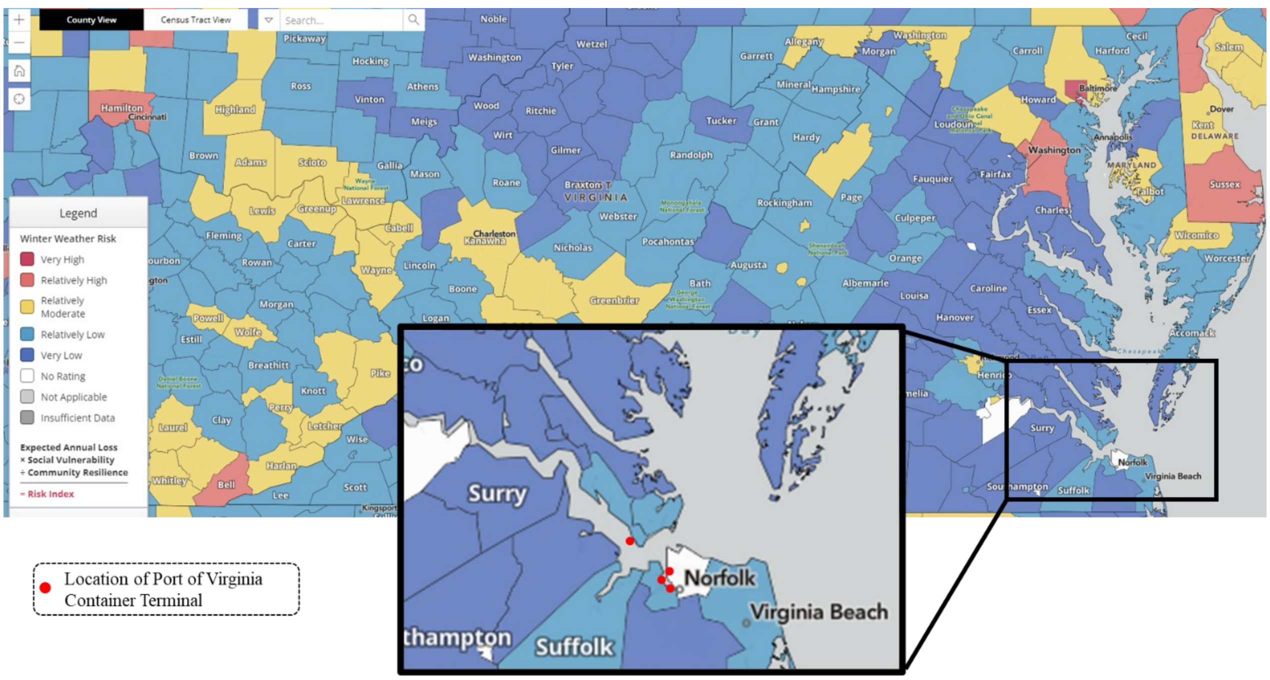

Figure 4-12. Map of relative winter weather risk, including estimated annual losses, social vulnerability, and community resilience. Adapted from (VDEM, Loose 2023)

Extreme cold is a cold weather event that lacks precipitation. The event may be acute, such as a wind chill advisory, but the greater threat is a long and sustained period of extreme cold. Definitions for how cold and how long an event must last differ across localities, even within Virginia, but a few days at or below 32 degrees Fahrenheit will typically meet the conditions.

Extreme cold has a severe impact on poor communities, both urban and rural. The Virginia HMP determined extreme cold was a *high* risk hazard due to the major risks to life. The Hampton Roads HMP did not assess the extreme cold hazard.

Extreme heat or a heat wave is a prolonged period of hotter than expected temperatures – typically defined as a weeks-long period with temperatures more than 10 degrees Fahrenheit over average for that time of year. The threat of this hazard increases in areas with high humidity. As with extreme cold the threat of this hazard is primarily one of public health. Heat exhaustion, heat stroke, chronic dehydration, and other heat related illnesses are common during an extreme heat event, often inundating or overloading medical systems. Furthermore, increased energy usage during periods of high heat may lead to black or brownouts, compounding the issue. The Hampton Roads area experiences high temperatures regularly – the maximum average high temperature for summer months is higher than 90 degrees Fahrenheit. Further, the Hampton Roads region is getting hotter on average – the hottest months on record since 1895 have occurred in the last three years (2019-2022). This poses a long-term threat to port operations, as increased temperatures hurt productivity and endanger workers. Figure 4-13 shows the high temperatures for Virginia, including specifically the Hampton Roads and Port of Virginia areas. The Virginia HMP and Hampton Roads HMP both determined that extreme heat was a *medium* risk.

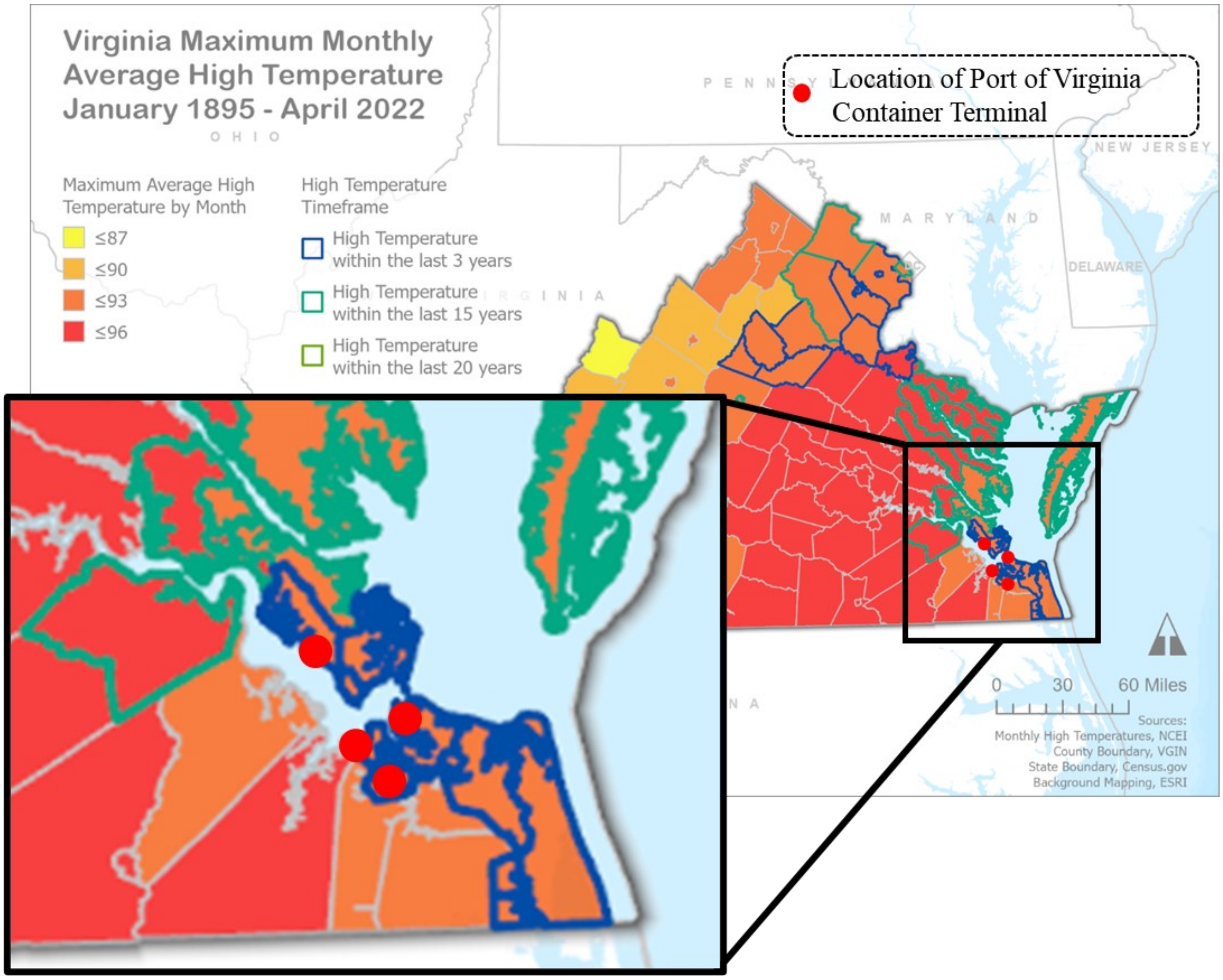

Figure 4-13. Maximum monthly average high temperatures for Virginia. Adapted from (VDEM, Loose 2023)

Erosion is the natural geologic process through which materials such as stone or soil are displaced by water, wind, or other phenomena. In the context of hazard analysis, erosion typically refers to *coastal erosion*, or the process through which beaches and banks are altered by weathering over time. Coastline erosion is a natural process, but it often accelerated by human activities, including the influence of climate change, rising sea levels, poor land use, and the destruction of protective measures. Erosion depends on a variety of factors such as the soil composition of the area, but all erosion is a threat to local infrastructure, including roads, bridges, and buildings. The opposite of erosion is *accretion*, the process by which new sediment is added to a region. Accretion is especially harmful to the Port of Virginia, as new sediment decreases the depth of the berths, leading to an increase in need for dredging projects, deepening the area around the port. Figure 4-14 shows the areas in Virginia with the greatest shoreline change over the last 100 years – note the black square in the southeast portion of Virginia, highlighting the areas around the Port of Virginia. Erosion elsewhere in the state, especially from the north in the Chesapeake Bay, leads to more sediment deposit in the area around the port. The statewide HMP determined that coastal erosion was a *medium* risk hazard. The Hampton Roads HMP did not assess coastal erosion.

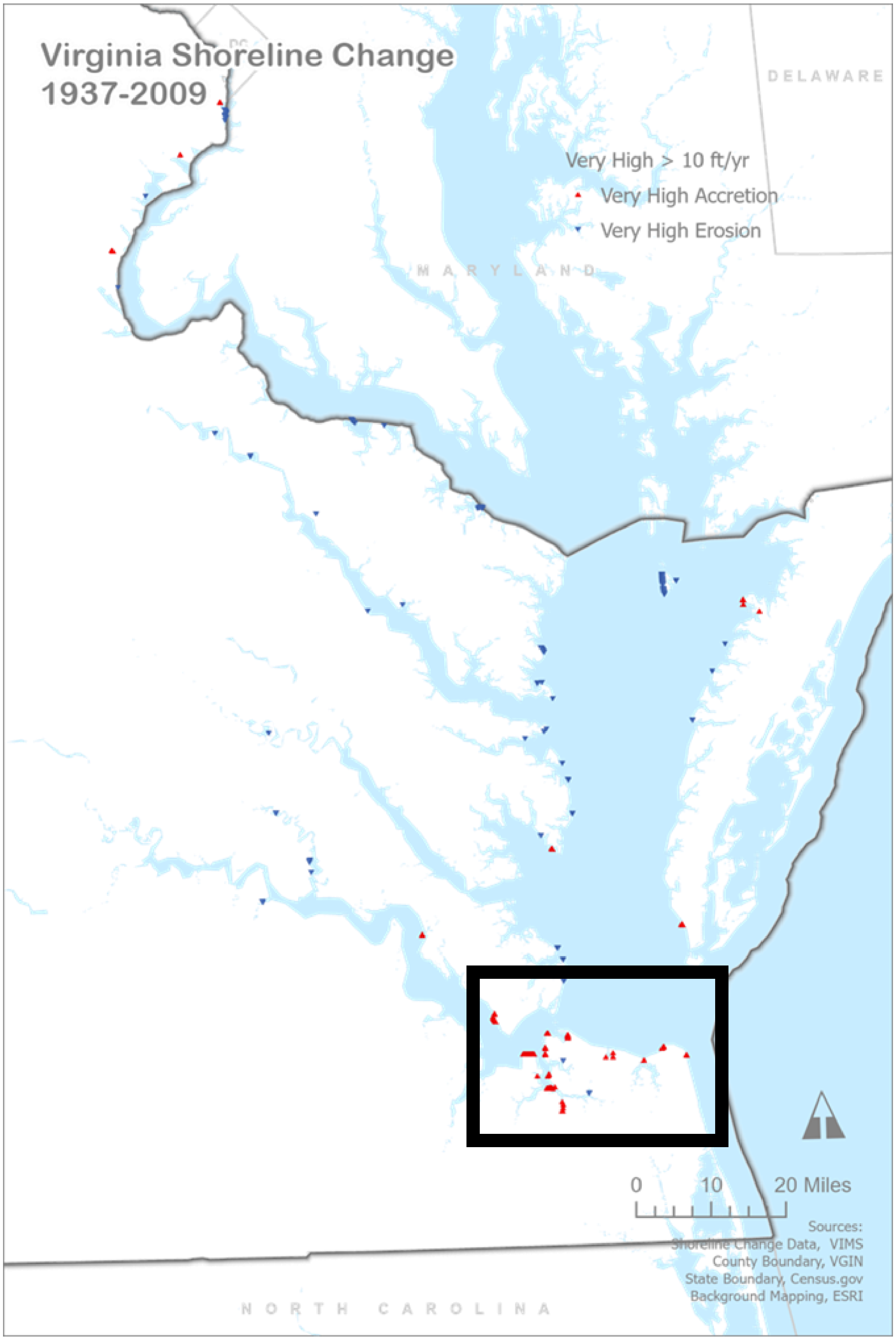

Figure 4-14. Map of shoreline change in Virginia, 1937 – 2009, highlighting the Port of Virginia and Hampton Roads region. Adapted from (VDEM, Loose 2023)

Wildfires are uncontrolled fires in natural environments such as grasslands, forests, or brush. Wildfires are typically started on accident by humans, but some are caused by lightning strikes or other natural occurrences. The severity of wildfires is heavily influenced by weather, with drought and high wind conditions leading to the rapid, uncontrolled spread of fire. The short-term impacts of wildfires include loss of property, including potentially severe impacts to local wildlife and vegetation. Loss of vegetation may lead to a loss of stabilization in soil, increasing the impacts of flooding and landslides. Wildfires are fairly common in Virginia, with about 700 occurrences per year on average, mostly in the mountains regions where steep slopes inhibit fire control measures. Wildfires thrive in rural communities which are often ill-equipped to handle the fires without state resources. Further, climate change leading to drier conditions in Virginia is expected to increase the frequency of wildfires. As such, the Virginia HMP ranks wildfires as *medium* risk. The Hampton Roads HMP ranks wildfires as *low* risk.

Communicable disease or a pandemic hazard is a widespread occurrence of an illness through an infectious agent such as bacteria, viruses, fungi, parasites, or prions. "Infectious" refers to the ability of an illness to survive and multiply within a host, while "infectiousness" refers to the ability of the infection agent to spread from host to host. Though infectious diseases can affect animal populations, this hazard refers to diseases that impact humans – however, animal-borne diseases are closely tracked to determine if they may mutate to affect humans. Infectious diseases include the COVID-19 SARS-CoV-2 virus, Zika virus, influenza, and Ebola, each of which has led to pandemic conditions. Pandemics are especially dangerous in poorer and more dense population areas. This includes parts of the Hampton Roads region. Figure 4-15 describes the CDC Social Vulnerability Map based on pandemic exposure. The blue regions are more susceptible to the impacts of pandemics. The Hampton Roads area is one of the most vulnerable regions in the

Commonwealth. The Virginia HMP ranked pandemics as *medium* risk hazards. The Hampton Roads HMP ranked pandemics as *low* risk hazards.

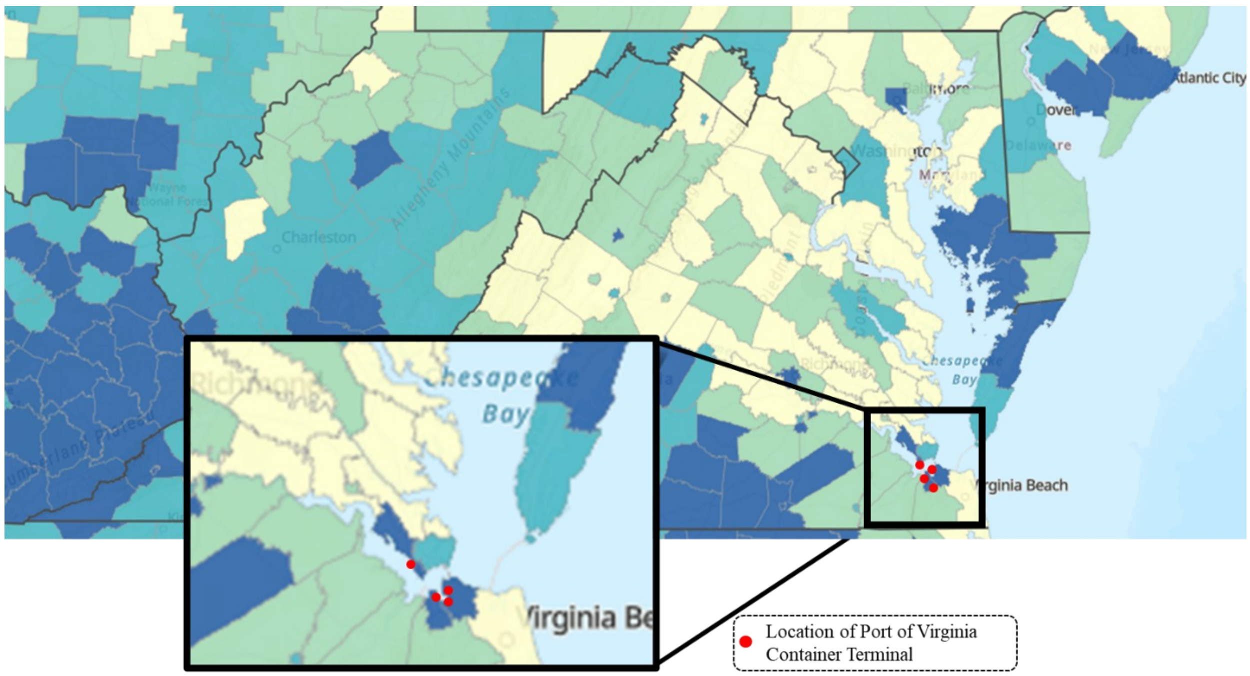

Figure 4-15. Map of the CDC Social Vulnerability Index – Pandemic Exposure for Virginia. Adapted from (VDEM, Loose 2023)

Tornados are windstorms characterized by a rotating funnel cloud extending to the ground. Tornados develop due to the interaction of cool, dry air and moist warm air, resulting in the twisting shape. Tornados produce high winds and can blow debris at high speeds – tornado wind speeds can range from 40 to 200 miles per hour. Most tornadoes are a few dozen feet in diameter and only touch down for a few minutes, but even smaller tornadoes are capable of massive damage. Tornadoes are ranked on the Enhanced Fujita scale, a 0-5 scale (EF0 lowest, EF5 highest), which indicates the top wind speed and likely damage caused by a tornado. EF2 tornadoes are exceedingly rare in Virginia, though have occurred before. Of particular interest to the Port of Virginia and Hampton Roads is the waterspout phenomenon – a tornado that forms over water.

Waterspouts are uncommon in Virginia, but do pose a special risk to container vessels and other maritime equipment. As such, both the Virginia HMP and Hampton Roads HMP consider tornadoes to be a *medium* hazard.

Drought is an extended period of limited rainfall within a large geographic area. The impacts of droughts are multifaceted. An extended period of drought can lead to water shortages that endangers the citizens of the region. Droughts also have economic impacts on farmers through increased irrigation coasts, increased feed costs, and impacts to animals. Wildlife may be impacted by the forced migration of local species. Fish and other aquatic wildlife may lose their habitats entirely. Drought combined with wind, high temperatures, and low humidity can lead to wildfires. There are four primary classifications of droughts – meteorological, agricultural, hydrological, and socio-economic. Meteorological droughts are due to abnormally high dryness over extended periods. Agricultural droughts are characterized by the specific impacts of low water levels – for example, water may be available, but not in the proper levels for the proper growth and development of crops. Hydrological drought is a shortage of groundwater supplies due to a lack of rainfall. Socio-economic droughts are caused by water shortages that limit water supplies of citizens for drinking and/or personal use. Both the statewide and Hampton Roads hazard analysis determined that droughts are a *medium* risk.

Hazardous materials (HAZMAT) incidents are the accidental or intentional release of chemicals or other materials that present a significant risk to public health, infrastructure, or the environment. HAZMAT incidents may last from a few hours to several days, while the impacts of the incident can be felt for longer periods. There are five primary classifications of HAZMAT incidents in the Virginia HMP – fixed site, waterway, highway, pipeline, and railway. All five of these may impact the Port of Virginia, though waterway, highway, and railway may impact operations more directly. There are nine classes of HAZMAT – explosive, gas, flammable liquid, other flammable substance, oxidizing agent and organic peroxide, toxic and infection substance, radioactive substance, corrosive substance, and miscellaneous. HAZMAT incidents may pose immediate health risks or damage infrastructure at the port, and the fallout of an incident may impact future business. Figure 4-16 describes the major assets of eastern Virginia and their locations, including the HAZMAT buffer zones of railroads. There is a high density of critical assets and railroad buffer zones south and east of the Port of Virginia. For this reason, both the Virginia and Hampton Roads HMPs rank HAZMAT incidents as a *medium* risk hazard.

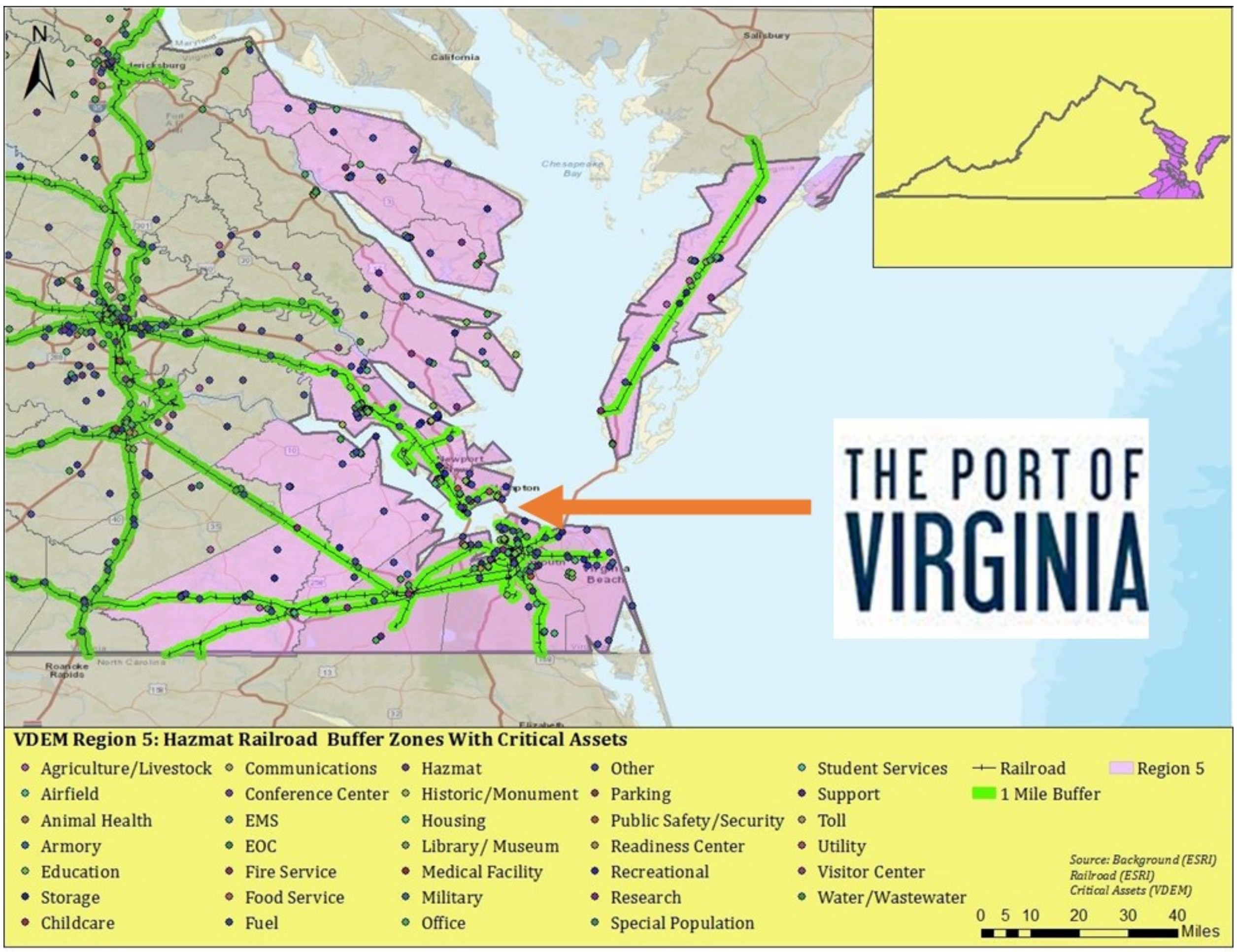

Figure 4-16. Hazardous materials incident buffers for rail with critical assets. Adapted from (VDEM, Loose 2023)

Landslides are a class of ground movements that includes the transport of soil, rock, mud, or debris, typically from an elevated position to a lower position. Landslides can be triggered by heavy rains, earthquakes, erosion, or human-caused disturbances. Landslides may occur quickly or slowly over extended periods. Landslides have the potential to destroy buildings, fracture roads or rail, destroy pipelines and water systems, and destroy power and communication lines. Different types of landslides have different effects. Rockfalls occur when large blocks of bedrock break from a cliff face. Rockslides occur when a mass of rocks slide down an inclined surface. Earthslides are the movement of soil across sheets of bedrock. Creep is the slow movement of earth over time. Debris flow is the development of a slurry of rock, earth, and human-constructed edifices due to an increase in water, falling down slopes. Landslides do occur in the Blue Ridge Mountain region of Virginia, but are typically small scale and occur far from urbanized areas. Due to the limited scope of landslide incidence and low risk to the population and infrastructure, the Virginia HMP determined that the landslide hazard was *low* risk. As the region around Hampton Roads and the Port of Virginia is a coastal plain with little slope, the Hampton Roads HMP did not assess the landslides hazard.

Land subsidence is the vertical movement of land either quickly or over extended periods of time, differentiated from landslides by the lack of sloping surfaces. Sinkholes are a type of land subsidence. Subsidence is frequently caused by human activity such as groundwater removal, oil pumping, or the removal of natural gas or minerals. Subsidence can also be caused by natural events, such as a karst – the collapse of land due to the erosion of underground soluble rocks. Land subsidence that occurs in populated areas can damage infrastructure. Subsidence can also occur slowly over time, as it has in the coastal region of Virginia. As the land sinks, sea level rises and threaten coastal communities. Land subsidence is not widespread across Virginia, so the statewide

HMP ranks it as a *low* risk hazard. However, due to its coastal location and the exceptionally high extraction of groundwater, the Hampton Roads HMP ranks land subsidence as a *high* risk hazard. Figure 4-17 describes the groundwater level decline due to groundwater extraction in the Hampton Roads region. The Port of Virginia falls within the 35-to-40-meter groundwater decline level.

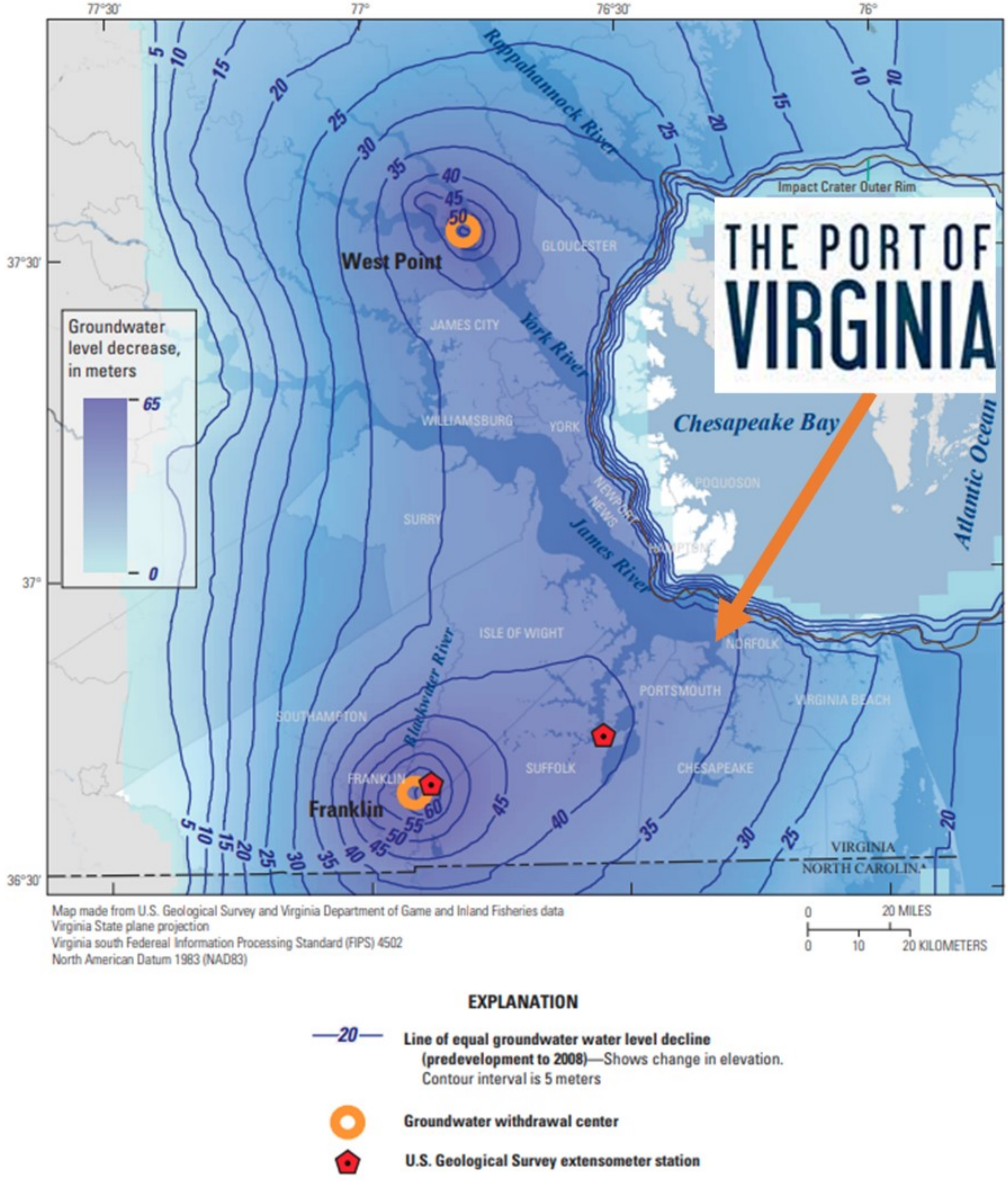

Figure 4-17. Map of the aquifer-system compaction caused by the withdrawal of groundwater. Adapted from (VDEM, Loose 2023)

Earthquakes are the shaking of the ground caused by sudden changes to the Earth's crust, resulting in seismic waves. Earthquakes can impact very large areas at once, potentially resulting in high levels of loss, mostly through the damage or destruction of buildings. Earthquakes vary in magnitude, with small earthquakes occurring more frequently than large earthquakes. Earthquakes are traditionally measured by the Richter Scale, the logarithm of the amplitude of seismic waves caused by the ground movement. More recent research indicates that assessing earthquakes by the intensity of effects on people and structures on the surface is a more reliable scale. This method more adequately describes how damaging an earthquake is to the affected area. Virginia contains two seismic zones – the Central Virginia Seismic Zone and the Giles County Seismic Zone. There are single digit to tens of earthquakes in Virginia each year, the overwhelming majority of which have little influence on people and structures. The low frequency and intensity of Virginia earthquakes lead to the statewide and Hampton Road to a *low* risk rating.

Impoundment failure is the flooding and destruction caused by the collapse or breach of dams or levees. Impoundment failure can occur without warning, sometimes due to heavy rains, though often due to poor impoundment upkeep leading to a failure. Total impoundment failure may occur within a few minutes of the initial breach and are difficult to predict. Impoundment failures that occur near population centers pose a significant risk to public health, and may damage or destroy buildings and other infrastructure. Flooding into natural areas may cause severe environmental damage, felling trees and leveling habitats. Dams are rated has no, low, medium, and high hazard based on their size, condition, and location relative to population centers. Figure 4-18 describes the dams ranked as a high hazard threat in Virginia. There are 23 high hazard dams in the Hampton Roads region. Strict regulations and monitoring of these dams led to the Virginia and Hampton Roads HMPs rating impoundment failure as a *low* risk hazard.

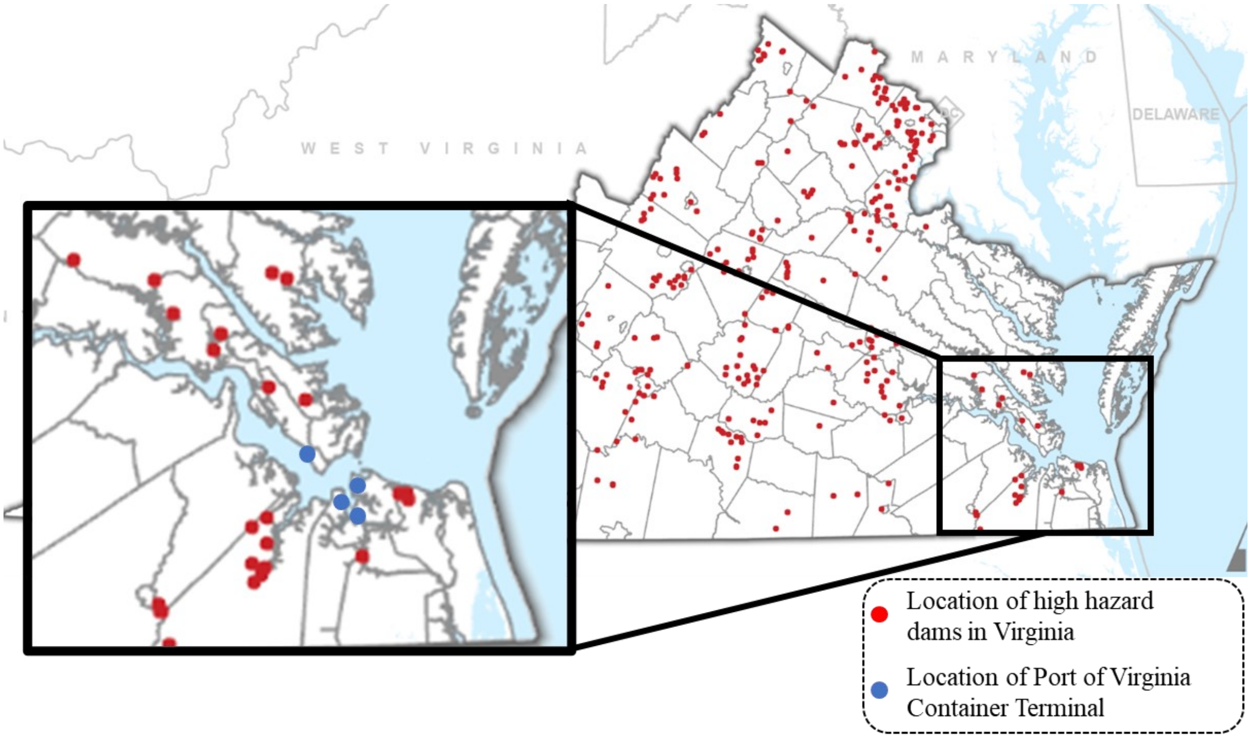

Figure 4-18. Map of the locations of high hazard dams in Virginia. Adapted from (VDEM, Loose 2023)

Terrorism and active threats include shooters, bomb threats, and other human caused complex coordinated attacks. These are acts that require the synchronized actions of individuals or teams across multiple locations that act with little warning and employing weapons to terrorize or harm the population (DHS 2018). Individual active shooters are also included in this category. This type of active threat poses immediate concerns to affected populations and has the potential to lead to infrastructure damage. These events are unpredictable. However, a low incidence of occurrence led Virginia HMP to list terrorism and active threats as a *low* risk hazard. Hampton Roads did not assess this hazard.

Space weather is a broad term used to describe any conditions that originate outside the Earth's atmosphere. There are three classes of space weather. Geomagnetic storms produce electrical currents that severely disrupt energy networks, causing blackouts and other failures. These are frequently long-term as equipment can become damaged. Geomagnetic storms may also disrupt GPS networks. Solar radiation storms occur high in the Earth's atmosphere – high levels of radiation exposure can cause illness or be deadly. Special precautions are taken to prevent high doses of radiation for aircraft operators and passengers operating in these regions at the relevant altitudes. Radio blackouts can also occur, impacting high frequency communications. The Virginia HMP determined that space weather is a *low* risk threat. The Hampton Roads HMP did not assess space weather.

4.5 Gap analysis

This section describes the gap analysis portion of the risk assessment. The purpose of a gap analysis is to highlight deficiencies in a system relative to the desired state. This is accomplished by assessing the current status, establishing expectations for performance, identifying gaps between current status and expectations, and identifying a set of recommendations for closing gaps. In the context of the Virginia and Hampton Roads HMPs, the gap analysis identifies existing resilience techniques for addressing the impacts of hazards outlined in the previous section. Resilience techniques are methods utilizing teams of people, processes, technology, and structures that can reduce the impact of or hasten recovery from a disruptive hazard event. For example, the Flood Mitigation Assistance Grant program administered by FEMA is a resilience technique that reduces the impact of flooding. This technique addresses multiple hazards such as hurricanes, flooding, and dam failure. Once the current set of capabilities are cataloged, the blue team identifies a desired future state of resilience techniques. The difference between existing and desired capabilities are called *gaps*. Figure 4-19 describes the gap analysis process utilized by the blue team.

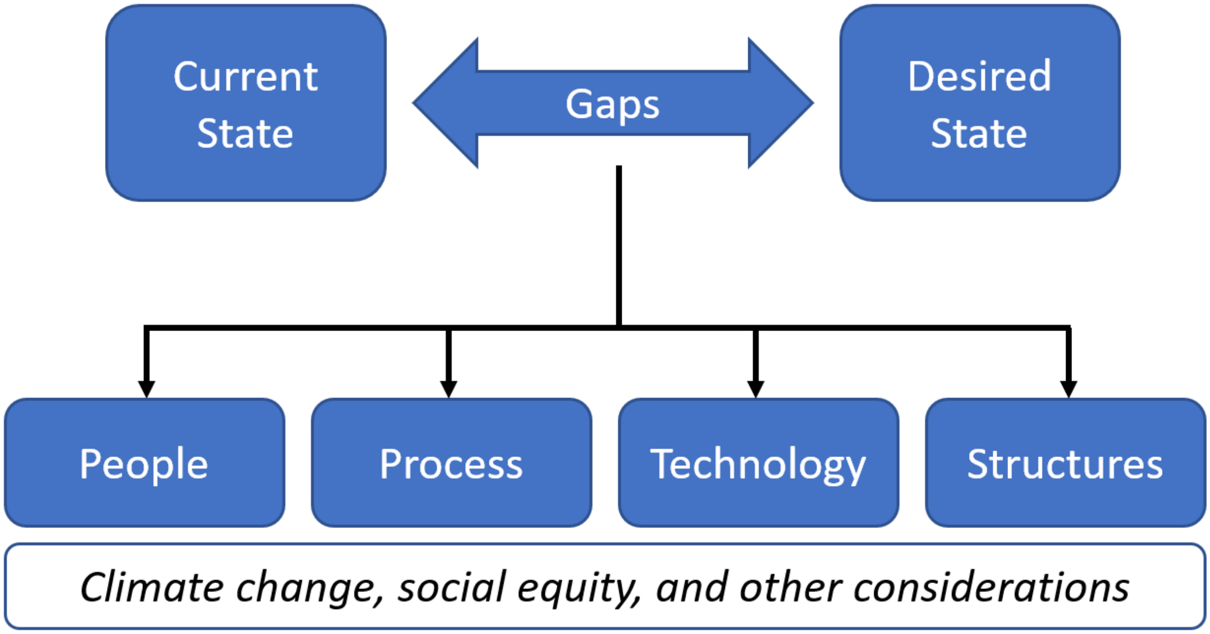

Figure 4-19. Gap analysis process for identifying hazard resilience techniques. The current and desired state are assessed for people, processes, technology, and structures that can reduce the harm caused by a hazard. Techniques that address climate change and social equity are of particular interest.

The first stage of the gap analysis is to establish categories – in this instance the categories are a set of hazards, or potential events that would harm the operations of the port. The next stage is to identify current mitigation capabilities of the Port of Virginia, Hampton Roads, and Virginia as a whole for each hazard. The Virginia Port Authority and other agencies across the state release reports detailing existing mitigation strategies (VDEM 2018). These reports are used to generate the current state. To create a vision for the future state, the blue team looks to federal guidelines, other ports, or other states to identify state-of-the-art mitigation strategies and resilience techniques, as well as developing new in-house strategies. Differences between current capabilities and the state-of-the-art are noted as gaps. Once gaps are identified the blue team generates recommendations for remediation. This may include adopting new strategies or finding other ways to address gaps.

Gaps are assessed across four primary dimensions including (i) the people and teams who execute mitigation actions such as the Silver Jackets and fire marshals, (ii) the processes in place for reducing harm during and after hazards such as the Virginia Know Your Zone hurricane program, (iii) mitigating technologies such as a hardened electric grid and model-based forecasting, and (iv) structures used to support individuals and organizations such as flood insurance.

The blue team also determines how hazards impact the Port of Virginia. A single new resilience technique may address multiple gaps if the hazards have the same impact on system performance. This dissertation focuses on hazards that impact the arrival of vessels and trucks to the port. For example, a hurricane, winter weather, and tornadoes may all have the similar effect on the port of reducing truck traffic during and immediately after the event. Once conditions return to normal, truck traffic may increase rapidly as drivers return to work at the same time leading to congestion. The results of the system description and gap analysis are supplied as artifacts to the T2 and red team as part of stages three, four, and five in the CPSRM.

The primary artifact of the gap analysis is a set of potential new resilience techniques that are given to T2 for implementation in stage three of the CPSRM. These capabilities should have a description, an origin, a list of the hazards addressed, a lead agency, a status, and a further comments and details. Table 4-4 provides an example of a new resilience technique uncovered by the gap analysis for the Port of Virginia. Appendix A.2 Gaps contains the complete set of proposed resilience techniques.

Table 4-4. Example of a new resilience technique developed as part of the gap analysis. This particular technique is specific to the Port of Virginia. The Port of Los Angeles has developed several models and simulations as part of standard operations. This identified gap develops a unique solution to the container stacking problem as outlined in this dissertation.

Machine Learning Enhanced Container Handling	
Description:	A digital solution for addressing known bottlenecks in the container handling process, especially the container stacking problem in container yards. The software utilizes machine learning and optimization to forecast demand and predict truck turn times. These results are shared with trucks, assisting with planning. The program has shown promising results, improving productivity and efficiency, reducing total emissions. While many ports utilize data and machine learning as part of operations, many processes remain black box and rules based. This resilience technique proposes a simulation and reinforcement learning based solution, specific to the Port of Virginia. In particular, the MuZero RL algorithm is applied due to its plan ahead capabilities and other factors.
Origin:	Port of Los Angeles
Hazard(s) Addressed:	Flooding, hurricane, winter weather, tornado
Lead Agency:	Virginia Port Authority
Status:	Planning, Ongoing
Comments:	The proposed resilience technique utilizes reinforcement learning and simulation due to data constraints. The reinforcement learning algorithm can be used as a companion to the controller for the rail mounted gantry cranes in a container stacking block.
Further Details: **https://www.portoflosangeles.org/business/supply-chain/port-optimizer%E2%84%A2** **https://www.prnewswire.com/news-releases/quantum-computing-application-sees-real-world-success-at-pier-300-at-the-port-of-los-angeles-301455106.html**	

Chapter 5: Development of system simulation

5.1 Overview

This chapter describes the simulation of the container stacking problem as outlined in previous chapters. The simulation is used for the development and testing of reinforcement learning algorithms to act as controllers for the container stacking process. Section 5.2 describes background information for the development of the simulation. Section 5.3 describes the development of the simulation in the Gym environment, including Section 5.3.1 outlining the components of the simulation, Section 5.3.2 describing the *Container* class, Section 5.3.3 describing the *ContainerStack* class, and Section 5.3.4 describing the *GymStackEnvironment* class. Section 5.4 describes a sample of results and performance of the simulation.

This chapter presents stages three, four, and five of the CPSRM. In stage *S3 – Develop Solutions*, T2 utilizes inputs from the blue team regarding the risk assessment to form a model of the target system, in this case a simulation model. In stage *S4 – Assess Solutions*, the red team implements and tests capabilities for the for simulation. Based on the feedback of the red team, T2 may adjust the resilience solutions. In stage *S5 – Revise System Description*, the updated system requirements based on T2 and red team feedback are submitted to the blue team for approval. Figure 5-1 describes the relationship between the content of this chapter and the CPSRM.

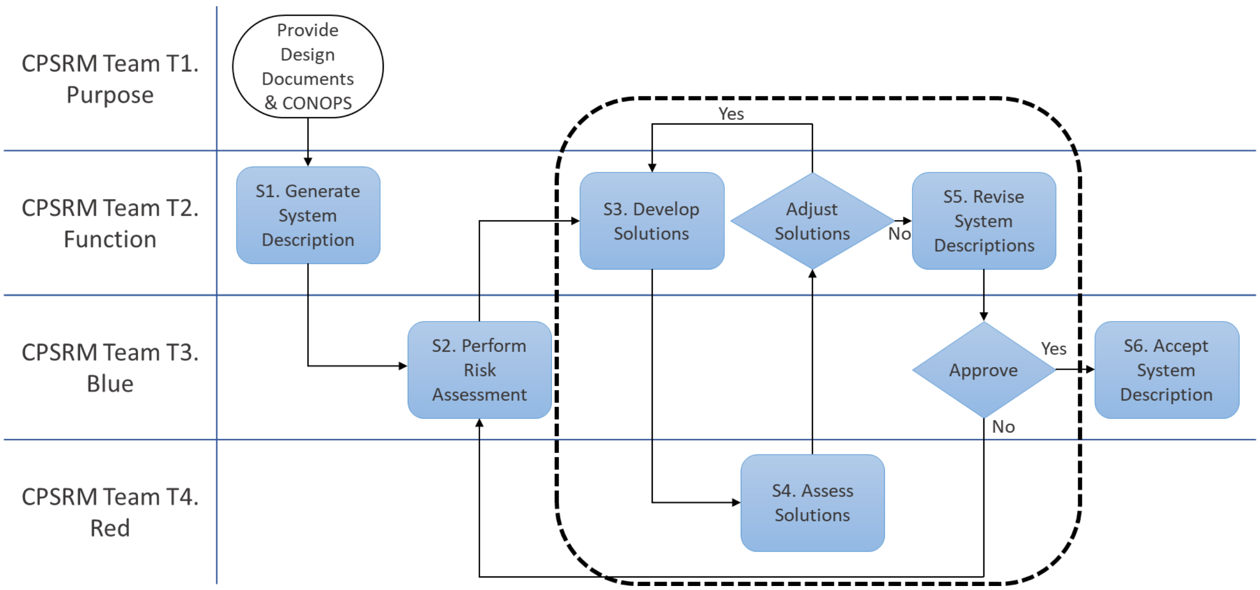

Figure 5-1. Relationship between CPSRM stages three, four, and five, and Chapter 5 – Development of system simulation. Chapter 5 describes stages three, four, and five of the CPSRM, in which T2 develops a simulation model of the target system to address the operational risks provided from stage two. The red team validates and tests resilience techniques, such reinforcement learning-based controllers.

5.2 Background

This chapter develops a high-fidelity simulation model than can be used to test reinforcement learning solutions to the container stacking problem. A *high-fidelity* simulation is a model with a high degree of realism or exactness in parameters and design. The simulation presented in this research follows the physical mechanics of a container stacking block, allowing for the integration of various control algorithms for planning and resilience in ports. The container stacking blocks are a known source of process inefficiency, forcing trucks to idle while they await containers that are stacked under one or more other containers. Further, each container move requires energy to operate the cranes and a human-in-the-loop to ensure operations are not harmful to the container or trucks. As such, minimizing moves is an important cost saving and environment sustaining measure. The development of a simulation provides two benefits. First, it allows the port to set a baseline performance and analyze operations without interacting with real containers or RMGs. This includes testing the impacts of disruptions as well as testing new operational techniques. The second benefit is that the simulation enables the training of reinforcement learning models.

Simulation is an effective tool for and testing different inputs, parameter changes, noise, and alterations to systems due to disruptive scenarios (Blum *et al.* 2021, Carter 2018). Previous efforts have used simulation to predict and plan port activities – for example, to assign vessels to berths and for fleet management (Almutairi 2017, 2016). Modeling and simulating port operations is a major area of new research. Improving operations with simulations reduces the environmental impact of the port, reduces wait times, and saves energy (Port of Los Angeles 2022). Simulation for various aspects of the container handling process is a subject of active research. Some research focuses on reducing truck turn times (Bosch 2022, Fotuhi *et al.* 2013). Other research focuses on the allocation of container slots to reduce the total number of container moves (Sikorra *et al.* 2021). Industrial software suites are available for simulating port processes as a whole, and container handling specifically (Simio 2022).

The simulation presented in this dissertation advances the existing research of container handling in two respects. First, this dissertation develops a simulation using the Gym framework developing a container stacking block is already populated by containers when the simulation begins. Second, this dissertation presents a simulation environment that enables the agent to choose *inaction* as a valid activity. This simulation has more time steps than moves required to place, reshuffle, and deliver all containers to trucks. This flexibility in the simulation allows the reinforcement learning algorithm to explore opportunities to reshuffle the container stack during downtime, ultimately reducing the number of touches per container.

5.3 Overview of simulation

This section provides an overview of the simulation and its development. Section 5.3.1 provides a summary of relevant details for the development of the simulation. Section 5.3.2 describes the *Container* class of the simulation. Section 5.3.3 provides a description of the *ContainerStack* class. Section 5.3.4 provides a description of *GymStackEnvironment* class.

5.3.1 Simulation components

The system specification outlined in Chapter 4 provides the requirements necessary for the development of a system simulation. This section describes the formation of the simulation, including environmental parameters, available actions, and other critical elements of the model. Figure 5-2 provides the high-level overview of the activity of simulated environment.

There are three primary elements of the simulated environment – the arriving containers (in-containers), the container stacking block, and the departing containers (out-containers). There is one agent in the simulation – the controller that determines which containers should move where and at what time. The in-containers arrive and are placed into a valid location within the container stacking block by the agent. Within the container stacking block the agent has the ability to move containers between stacks at any time. The out-containers are removed from the container stacking block and delivered to trucks, exiting the simulation.

Many confounding factors are abstracted out of the simulation. For example, two RMG cranes service each container stacking block in the Port of Virginia. The simulation removes this nuance, sending only instructions on which containers move and to which locations rather than on which crane would perform which activity. Another factor is the element of time – that is, it takes more time for a container to traverse the length of the entire stack than half of the stack, altering the wait time of a truck. These factors have limited relevance on the ultimate performance metric – the minimization of container touches – and would increase the computational complexity of the reinforcement learning model if included.

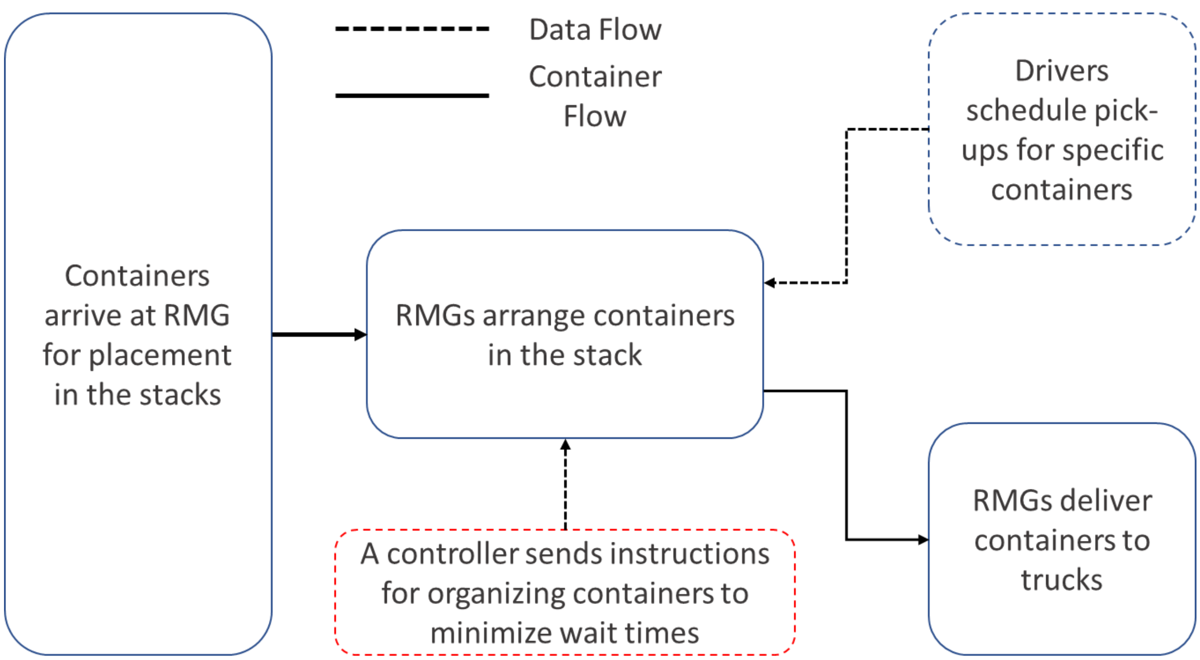

Figure 5-2. The high-level system specification for the container handling process for a single container stacking block. In this simulation, containers arrive to the stack, receive an initial placement, are rearranged as needed, and are delivered to a truck.

Figure 5-3 shows a graphical representation of a container stack. The figure defines critical elements of the simulation environment. A “stack” is a single set of containers oriented vertically, placed one on top of the other. A stack may be empty with zero containers, up to the maximum allowable height of a stack (in Figure 5-3, the maximum height is four). A “bay” is a set of stacks oriented crosswise, and is limited by the width of the RMGs servicing the block. A “lane” is a set of stacks oriented lengthwise from the entry point to the exit point of the container stacking block. A “tier” is a set of containers (or container slots) that corresponds with the height of a stack. The maximum number of tiers in a stack is limited by the height of the RMG and health and safety protocols.

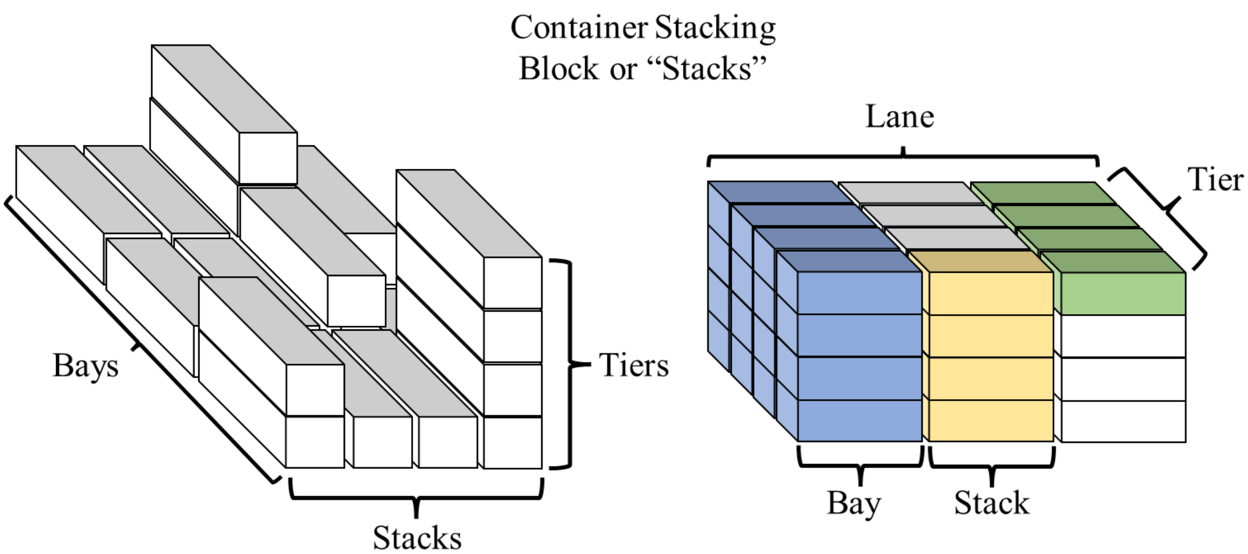

Figure 5-3. Graphical representation of a container stacking block. Adapted from (Guven and Eliiyi 2014)

Various ports manage the container stacking problem in different ways – these are largely heuristic in nature and are trade secrets (Pascual *et al.* 2016). The container stacking problem is difficult to resolve due to substantial uncertainty in the movement of containers when outside of the port (vessel arrival) and when containers will leave the port (truck departures). This uncertainty is captured as the *dwell time* of a container.

Dwell time is the distribution of time that a given container may remain in the stacking block. Current port operations utilize dwell time in decision making – a container that is likely to remain in the stacks for several more days is likely to be placed under containers with shorter remaining dwell times. However, the uncertainty of when a container will leave the stacks strains effective decision making. This is further exacerbated by disruptive events such as winter weather or thunderstorms – the departure of containers changes in unpredictable ways. The simulation of the container stack utilizes this information to develop a model that closely reflects actual stacking operations. Figure 5-4 describes container dwell time distributions. Actual dwell times vary based on a number of factors including freight volumes, local demand, weather, and other confounding factors. However, research indicates that the dwell time of containers follows a normal distribution in aggregate (Hassan and Gurning 2020). Dwell time is an input to the simulation model based on existing Understanding the dwell time of containers is critical for the development of an accurate simulation.

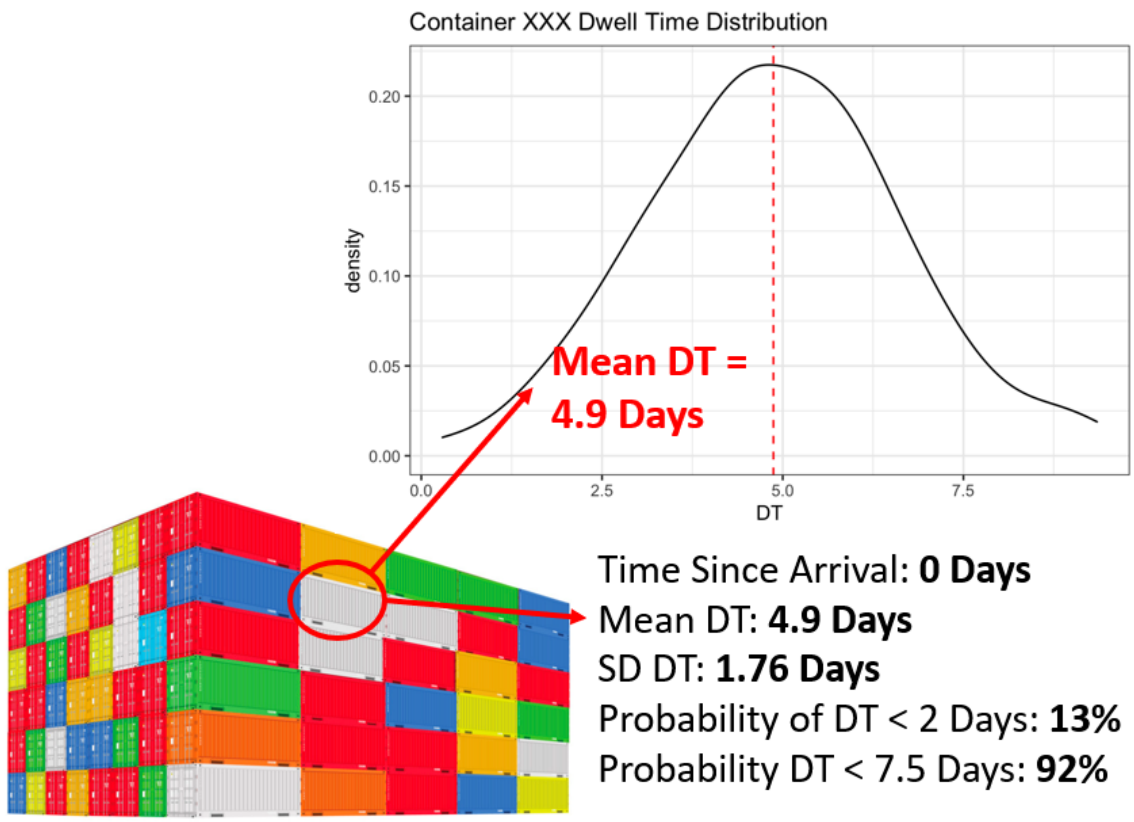

Figure 5-4. Container dwell time distributions. Ideally containers require two moves – one to place the container in the stack, and again for placement on a truck. Usually, three or more moves are required to reach the appropriate container. This is exacerbated in disruptive conditions (Hamdy, Loose *et al.* 2022).

The simulation is implemented in the Python programming language and leverages the Gym simulation environment. There are three primary programming elements of the simulation:

1. The *Container* class – the class for creating a container object. A *container* object includes several properties such as arrival time, dwell time distribution parameters, departure time, and others. The *container* class contains the logic required to sample a container from arrival and dwell time distributions and prepare the container for placement in the stack.

2. The *ContainerStack* class – the class for creating a container stacking block. A *ContainerStack* object includes several properties such as the number of stacks, tiers, and bays and the number of arriving containers. The *ContainerStack* class generates a set of containers that exist within the block before the simulation begins. A *ContainerStack* object provides important functions for manipulating containers in the block, including instructions for moving containers, determining which containers need to move, calculating legal moves, and visualizing the environment.
3. The *GymStackEnvironment* class – an implementation of the Gym environment for integration with reinforcement learning algorithms. The class contains all of the semantics necessary for performing a simulation. A *GymStackEnvironment* object includes several properties, including the observation space, the action space, and system time. The *GymStackEnvironment* also includes the core functions for running the simulation, including the *step* function, the set of legal actions, the render function, and the collection and delivery of other relevant metadata.

5.3.2 The *Container* Class

The *Container* class is used to create container objects. Figure 5-5 describes the pseudocode for the *Container* class. The full code can be found in Appendix A.3. A container object provides several parameters necessary for the development of the simulation. The class takes ID number, simulation run time, and the mean and standard deviation values of the dwell time distribution.

The ID number of the container is user-provided at the time of generation. The following section details how ID numbers are created, but the ID variable is represented by an integer. The simulation run time is listed in minutes and represented by an integer. The length of the simulation can vary based on testing and training needs. The length of the simulation also impacts the arrival rate of containers entering the system. The parameters for calculating the dwell time of the containers are drawn from existing research and based on real-world performance and both represented as integers (Hassan and Gurning 2020). For example, in a simulation equivalent of four weeks of operation, the distribution for dwell time is drawn from *N(1200, 306)* using minutes during operating hours as the unit, or roughly a 2.5 day average dwell time.

The class initializes three variables. First, the coordinates of the container as a [stack, tier] paired list of integers (initially [0,0] until placed). The coordinates correspond with the location of container in the stacking block. The coordinates will change as the container is moved. The container stacking block environment is represented as a Python list, whose first and last elements are reserved for arriving containers and departing containers respectively. The second variable is a Boolean flag indicating if a container begins the simulation in the container stacking block or is awaiting arrival. This influences how arrival and departure times are generated – generally a container that begins in the stacking block will have an arrival time before t = 0 and an earlier departure time than containers that have not yet arrived. The third variable is a Boolean flag indicating if a container is scheduled to exit the simulation. The flags are used in the Gym environment to determine which containers need to move and when.

Upon creation of a *Container* object, the class generates an *arrival time* sampled from a uniform distribution based on the length of the simulation. That is, the simulation assumes that containers are equally likely to arrive at any point during the simulation time. To ensure most of the arriving containers depart the simulation during the simulation time, the arrival time may be compressed to the first 80% of the simulation. *Arrival times* are represented by integers corresponding to system time. The class also has the capability to change the arrival rate to a triangle or other distribution to simulate disruptions. Containers that begin the simulation in the container stacking block can have arrival times from times less than zero, indicating that they arrived in the “past”. Once all containers have an arrival time, the dwell times are created. A noise term samples from a normal distribution with mean and standard deviation from the class parameters to generate the dwell time. This term is added to the arrival time such that *arrival time + dwell time = departure time.* The class then checks to ensure that all departure times occur after simulation time $t = 0$, regenerating departure times if this check fails.

There are three class methods. *setPosition* sets the coordinates of a container, corresponding to its location in the stacking block. *setArrival* sets the *arrival time* of a container based on a user provided argument. *setDeparture* sets the *departure time* of the container based on a user provided argument. Table 5-1 describes the variables of the *Container* class. Table 5-2 describes the methods of the *Container* class.

```
class Container:

    # ID -                          Generated ID for Container object
    # DwellTimeMean -               Mean dwell time for container object
    # DwellTimeStandardDeviation -  Standard deviation for container dwell time
    # RunTime -                     Length of simulation
    # InBaseStack -                 Is the container is initialized in the block?
    def __init__(ID, DwellTimeMean, DwellTimeStandardDeviation, RunTime, InBaseStack):

        # Declare variables
        Initialize variable for coordinates of container
        Initialize variable for if the container has arrived to the block
        Initialize variable for if the container is ready to depart

        # Sets the in-simulation arrival time of the container
        if the container is in the base stack:
            set arrival time to the past/near future sampled from uniform distribution

        else:
            set arrival time according to uniform distribution based on RunTime

        # Sets the dwell time of the container
        Initialize a noise term using the DwellTimeMean and DwellTimeStandardDeviation

        # Ensure the noise term is positive whole number
        # Containers cannot depart before they arrive
        while noise is less than 0:
            regenerate noise term

        # Set the departure time
        Add arrival time and the noise term to generate departure

        # Ensure the departure time is non-negative
        # This only applies to containers in the base stack
        while departure time is less than 0:
            Regenerate departure time

        #Class methods
        #------------------------------------------------------------------------------

        # Method to set the container coordinates
        def setPosition(stack, tier):
            Set coordinates to [stack, tier]

        # Method to set the container arrival time
        def setArrival(arrival):
            Set arrival time to arrival argument

        # Method to set the container departure time
        def setDeparture(departure):
            Set departure time to departure argument
```

Figure 5-5. Pseudocode for the *Container* class

Table 5-1. Overview of the key variables in the *Container* class

Variable	Data Type	Units	Description
ID	Integer	None	Identification number for individual containers. Used primarily in visualization of the environment.
Arrival Time	Integer	Minutes	The arrival time of the container listed as minutes after the beginning of the simulation, t=0. Used to determine when a container needs to be placed in the container stacking block and in the calculation of departure time.
Mean	Integer	Minutes	The mean value parameter of the container dwell time distribution. This value is used to calculate the departure time of a container.
Standard Deviation	Integer	Minutes	The standard deviation parameter of the container dwell time distribution. This value is used to calculate the departure time of a container.
Dwell Time	Integer	Minutes	The amount of time the container waits to depart the container stacking block, listed in minutes after the beginning of the simulation, t=0. It is sampled from a normal distribution using the mean and standard deviation parameters.
Departure Time	Integer	Minutes	The departure time of the container listed as minutes after the beginning of the simulation, t=0. Used to determine when a container needs to depart the container stacking block. Is calculated as *arrival time + dwell time.*
Max Run Time	Integer	Minutes	The maximum run time of the simulation, discretized into minutes. This value is used to create the arrival times and arrival distribution of containers entering the container stacking block.
Position Coordinates	List	None	The [stack, tier] coordinates of the container. This variable is initialized to [0,0] on creation of the container. The value is edited when the container enters or moves in the container stacking block environment.
Arrival Flag	Boolean	None	Boolean flag indicating if the container is eligible to be placed in the container stacking block. It is True if the arrival time is greater than or equal to the system time.
Departure Flag	Boolean	None	Boolean flag indicating if the container is eligible to be removed from the container stacking block. It is True if the departure time is greater than or equal to the system time.

Table 5-2. Overview of the methods in the *Container* class

Method	Parameters	Returns	Description
setPostition	Two integers	None	Used to update the position coordinates variable of an individual container. Called in subsequent classes to update the container position in the container stacking block, indicate that a container has left the simulation, and determine legal moves.
setArrival	Integer	None	Used to update the arrival time of a container to the parameter provided. This function is used in subsequent classes when resetting a container stacking block, restarting the simulation, or generating new containers
setDeparture	Integer	None	Used to update the departure time of a container to the parameter provided. This function is used for ensuring departure times are realistic (i.e., that a container does not depart before it arrives) and do not break the simulation.

5.3.3 The *ContainerStack* Class

The *ContainerStack* class is used to create a container stacking block object, including arriving and departing containers. Figures 5-6, 5-7, and 5-8 describe the pseudocode for the *ContainerStack* class. Appendix A.3 describes the full code of the *ContainerStack* class. A *ContainerStack* object contains the variables and methods necessary for the simulation of a single container stacking block. As all container moves are considered equally and move distance is not part of performance metrics for the reinforcement learning algorithm, the container stack is represented in two dimensions only – length (in number of stacks) and height (in number of tiers).

```
class ContainerStack:

    # Length -            The number of stacks in the environment
    # Tiers       -       The maximum number of tiers in a single stack
    # NumberIncoming -    The number of containers arriving
    # Runtime -           Length of simulation
    def __init__(Length, Tiers, NumberIncoming, Runtime, Seed):

        # Declare variables
        Initialize variable container ID to 0
        Initialize the environment as a list

        # Generate the initial environment
        # The number of stacks is equal to Length-2
        # Elements 0 and -1 are used for storing new arrivals and departures

        for the length of the environment:
            create a new stack
            for the height of each stack:
                # Set a utilization parameter (ex. 0.5 or 50%)
                Randomly generate containers based on utilization
                Set container ID, position [stack, tier]
                Place container in the ascribed position

        # Generate the incoming containers
        for the length of NumberIncoming:
            Randomly generate a container
            Set container ID, position in element 0 of the environment

        # Set departure times for each container
        for each container in the environment:
            Set the departure time using dwell time parameters
```

Figure 5-6. Pseudocode for variable declaration, initialization, and generation for the *ContainerStack* class

The *ContainerStack* class requires five parameters. The Length parameter is an integer indicating the number of stacks in the simulation. The first and last elements of the environment are reserved for the set of incoming and outgoing containers, so the total number of stacks in the container stacking block is equal to *Length – 2.* The Tiers parameter is an integer that indicates the maximum height of each stack. The Runtime parameter is an integer that indicates the maximum run time of the simulation in minutes. The NumberIncoming parameter is an integer that indicates the number of containers that will arrive during the simulation. The Seed parameter is used by the Gym environment to reset and repopulate the entire environment to a new, random state.

The *ContainerStack* class initializes two variables. The counter for the ID number of containers is set to 0. The empty list that will become a list of lists representing the container stacking block, incoming containers, and outgaining containers is initialized.

The class also generates three elements of the environment: the containers in the container stacking block, the arriving containers, and a check on departure times of all containers. The container stacking block creates a list of lists, where the inner lists represent the height of stacks and stores arriving and departing containers. The outer list contains a number of elements equal to the Length parameter. To populate the initial container stacking block, a random floating point value samples from a uniform distribution between 0 and 1 for each container slot. If the value is greater than or equal to a user defined threshold, a container is generated in that position in the stack. For example, if the threshold were set to 0.5, the initial container stacking block would be roughly 50% filled as around half of the container slots would have a random value of greater than 0.5. That is, if there are 50 container slots, roughly 25 slots will be filled at the start of the simulation. One open container slot is always reserved and left empty at the beginning of the simulation as part of error handling in the Gym environment.

The *ContainerStack* class then generates the incoming containers. The number of incoming containers is a user defined parameter. Incoming containers have an arrival time sampled from a uniform distribution beginning at $t=0$ and ending at the maximum run time of the simulation. Some iterations of the simulation, including the simulations in Chapter 6, use some fraction of the maximum run time to increase the proportion of arriving containers that will also depart during the simulation (for example, arrival times will be sampled uniformly between 0 and 0.8 of the maximum runtime). The simulation also provides the capability to sample from other distributions – for example a triangle distribution – to emulate the effects of an external disruption.

The number of arriving containers depends on the number of container slots in the container stacking block. At the Port of Virginia, during normal operations, a container stacking block will receive about 20 containers per hour, or one arrival every three minutes. However, Port of Virginia container stacking blocks are in an 8-bay x 5-tier x 30-lane configuration, or 1200 container slots. Several factors contribute to this arrival rate – available slots, the destination of the containers, expected dwell time, and the availability of other slots in other stacking blocks. Many of these elements are not included in the simulation. A benefit of the simulation is that users can increase or decrease arrivals and adjust arrival times to meet the current need.

Finally, the class checks the departure times of all containers in the environment, both arrivals and containers in the stacking block, to ensure they are compliant with environment rules (i.e., a container cannot be scheduled to depart before it arrives). This is helpful for the Gym environment which runs the simulation.

Figures 5-7 and 5-8 provide an overview of the *ContainerStack* class methods. The *resetStack* method is used to create a new random environment, including a new set of arrivals and prepopulated container stacking block. It is critical to note here that the *resetStack* method has the ability to preserve the "shape" of the initial container stacking block, including the number of containers in each stack and the arrival times of these containers. This feature is helpful for tuning the reinforcement learning algorithm, which relies on predictability when choosing actions in the environment.

```
#Class methods
#---------------------------------------------------------------------------

# Returns the environment to an initial state
def resetStack():
    Has new arrival times, departure times, and initial environment
    Used for reinforcement learning

# Moves containers according to the Action Space of the Gym Environment
def moveContainer(CurrentStack, DestinationStack):
    Select the container on top of the Current Stack
    Place the container on top of the Destination Stack

# Iterate over arriving containers and set flag for if they have arrived
def setArrived(time):
    if system time greater than or equal to arrival time:
        set arrival flag to true

# Iterate over the arriving containers and check for arrivals
def getArrived():
    Checks the arrival flag of the arrivals
    return list of containers that have arrived

# Iterate over environment to set departure flags
def setDepart(time):
    if system time greater than or equal to departure time:
        Set departure flag to true

# Iterate over environment to check departure flags
def getDepart():
    Checks the departure flags of containers in the environment
    return list of containers ready to depart

# Visualize the ContainerStack object
def showStack(factor):
    Print the list of lists, showing containers
        Factors include arrival time, departure, ID, position
```

Figure 5-7. Pseudocode for the first seven methods of the *ContainerStack* environment

The *moveContainer* method moves the top container of one stack to the top tier of another stack, taking these two integer values as parameters. The ability to move a specific container that may be placed below other containers is handled in the following section. The *moveContainer* method alters the location of containers within the list of lists, as well as updating the position coordinates of the container. No other values are changed.

The *setArrived* method iterates over all containers in the arrival element of the environment, checking if their arrival time parameter is greater than the system time. The method takes an integer as the time parameter. The arrival time of each container is compared to the time parameter of the method; if the provided time is greater than the arrival time of the container, the arrival flag of the container is set to True. The *getArrived* method iterates over all containers in the arrival element of the environment, checking the arrival flags. If the arrival flag is set to true, it is added to the list of containers that are ready for placement in the container stacking block. The method returns this list.

The *setDepart* method iterates over all containers in the container stacking block element of the environment, determining if the departure time parameter is greater than system time. The method takes an integer as the time parameter. This value is compared to the departure time of each container; if value is greater than or equal to the departure time, the departure flag of the container is set to True. The *getDepart* method iterates over all containers in the container stacking block, checking departure flags. All containers marked for departure are added to a list, which the *getDepart* method returns.

The *showStack* method is used to visualize the environment. All elements are printed to the terminal on individual rows, including the arriving containers, the stacking block, and the departed containers. The *showStack* method takes a String parameter, which corresponds to the parameter of interest for the visualization. For example, a user could enter "ID" when calling the method, which would display the ID numbers of containers, as well as their position within the environment. Examples of these visualizations can be seen in subsequent sections.

```
# Generates a dictionary object of what moves are possible
def Moves():
    Initialize counter

    for number of stacks in the environment:
        for number of stacks in the environment:
            #Generate pair indicating movement from first stack to second
            if first stack is the departures list:
                Pass
            else:
                generate the action pair
                Set dictionary key to counter, set value to action pair
                Increment the counter

    Set final action element to "pass"
    return dictionary of moves

# Return the set stack in the environment with open tiers
def validStackDestination():
    Initialize valid stacks to an empty list

    for stacks in the environment:
        if number of containers is < Tiers:
            add stack as valid destination

    return the list of valid destinations
```

Figure 5-8. Pseudocode for the final two methods of the *ContainerStack* environment

The *Moves* method is used to generate the set of all possible actions in the environment. There are three types of moves – moves from the arrival section to the stacking block, reshuffles that move containers from one stack to another within the block, and moves from the block to the exit node. The rules for which moves are legal at a given simulation time are defined in the *GymStackEnvironment*, so all possible pairwise moves are generated here. There are a few exceptions to reduce the size of the decision space and improve the performance of the reinforcement learning algorithm. For example, the *Moves* method does not generate moves with the same initial and final stack (ex. [0,0], [1,1], etc.). Though the reinforcement learning algorithm would eventually determine that this action is unhelpful, reducing the size of the action space improves training performance quickly. Other restrictions include eliminating the ability to move containers that have departed the system – that is, there are no valid moves from the exit node of the environment to the stacking block. The set of moves is stored as a Dictionary object, with {integer:list} as the key-value pair. The integers range from 0 to the maximum number of moves for the stack (for a stack of length 12, there are 110 moves). These each correspond with the two-element list of action pairs. After all action pairs are generated, a final move is appended to the list, "inaction", in which no containers move in the environment. As examples, a sample of actions are {0: [0,1]}, a move from the arrival node to stack one; {64: [6,4]}, a reshuffle from stack six to stack four and; {111: 'pass'}, a no move action.

The *validStackDestination* method is used to determine which stacks are available for placement of containers. That is, the method determines which stacks have fewer containers than specified by the Tiers parameter. The method iterates over all stacks in the stacking block and checks the current number of containers in the stack. If this number is less than the Tiers parameter, the stack location is added to a list of valid stacks.

Table 5-3. Overview of the key variables in the *ContainerStack* class

Variable	Data Type	Units	Description
Length	Integer	Stack	The total number of elements of the stacking block environment. The first and last elements of the environment are reserved for arrivals and departures, so the total number of stacks is equal to Length-2.
Tiers	Integer	Stack	The maximum height of an individual stack for the simulation.
Number Incoming	Integer	Container	The number of containers that enter the stacking block during the simulation.
Runtime	Integer	Minutes	The maximum runtime of the simulation, used to determine when containers arrive.
Seed	Integer	None	Used to change the randomly generated environment to new configurations.
ID	Integer	None	The counter for assigning ID numbers to containers upon generation in the environment.
Environment	List	None	The list of lists used to store arriving containers, containers in the stacking block, and departed containers. The containers are moved to new positions within the list of lists such that containers can be efficiently placed, added, and removed from the environment.

Table 5-4. Overview of the key methods in the *ContainerStack* class

Method	Parameters	Returns	Description
resetStack	None	None	Used to regenerate a new container stacking block environment, including new arrival and departure times, block utilization, and other factors. Has the ability to maintain the initial configuration of the stacking block and base stack arrival times for reinforcement learning.
moveContainer	Two integers	None	Used to move a container within the environment. This can be from the arrival queue to the stacking block, reshuffles within the stacking block, and from the block to the exit node.
setArrived	Integer	None	Used to set the arrival flag for containers within the arrival node to True if arrival team is less than or equal to system time.
getArrived	None	List	Used to generate a list of containers that need to be moved from the arrival node and into the stacking block.
setDepart	Integer	None	Used to set the departure flag for the containers within the stacking block to True if the departure time is less than or equal to system time.
getDepart	None	List	Used to generate a list of containers that need to be moved from the container stacking block to the exit node.
showStack	String	None	Used to print a visualization of the environment to the console. Can take any container parameters as input and display this information. Each stack is printed on its own line, mimicking the layout of the container stack.
Moves	None	Dict	Used to generate the dictionary of all possible legal moves, including from arrivals to the stacking block, within the block, and exiting the block. This is used as the action space for the Gym environment and to run the simulation.
validStack Destination	None	List	Used to generate the list of stacks within the stacking block that have open tiers.

5.3.4 The *GymStackEnvironment* Class

The *GymStackEnvironment* class is an implementation of the Gym environment. The Gym environment is a common framework for the development of simulation environments for testing reinforcement learning algorithms (Moerland *et al.* 2023, Brockman *et al.* 2016). Figures 5-9 and 5-10 describe the pseudocode for an implementation of the Gym environment for the container stacking problem.

The environment initializes five attributes – two are required by the Gym API and three are custom implementations. The Gym environment requires an "action space", which is the set of legal actions. These can be discrete actions (e.g., the direction a robot should travel in cardinal directions) or continuous (e.g., the direction a robot should travel in degrees from north). For this simulation, the action space is the dictionary of legal actions provided by the *Moves* method of the *ContainerStack* class that outlines the full set of possible actions. That is, the set of actions in the action space is a list of integers corresponding to the set of possible moves in the container stacking block environment. Gym also requires the an "observation space", or the representation of the simulated environment. Actions alter the observation space. For this simulation, the observation space is a randomly generated *ContainerStack* object with user defined parameters. The system time is initialized to zero (*t=0*) when a new *GymStackEnvironment* object is generated. The class also generates a move counter which tracks the number of container moves during the simulation to assist with performance assessments. The class initializes an empty list used to track which containers require a move at a given system time.

The *step* function is a core feature of Gym. The *step* function runs a single time step of the simulation utilizing the dynamics of the environment. The step function takes a single action as an input and provides a reward value, new information about the environment, and other metadata.

In this simulation, there are three types of step. The first type of *step* is a container move from the arrival bay into the container stack environment. As the arrivals do not follow the same system dynamics as the stacking block – for example, an arrival does not need to be “top picked” and can be moved from any position within the list – special logic is required to ensure the correct container is moved from the arrival bay to the stacks. The first type of step provides a reward of -1. The second type of step is any other container move – either between individual stacks or from a stack to the exit node. These moves do not specify which container needs to move, simply moving the top container of a stack to a new location. The second type of step provides a reward of -1. The third step is “inaction”, a step in which no containers move. The third type of step provides a reward of 0. Once the step is executed and the environment is altered, the time step increments by one, and the *step* function checks if the simulation is over. If there are no legal moves or if the simulation time has completely elapsed, the simulation is terminated. The *step* function returns the reward value of the single action, a flag for ending the simulation, and other metadata.

```
class GymStackEnvironment:

    def __init__():

        # Declare variables
        Initialize system time to 0
        Initialize a ContainerStack object as the observation space
        Initialize the action space as the set of all possible moves
        Initialize a counter for the number of container moves
        Initialize the set of containers that require a move

        #Functions for the Gym Stack environment
        #---------------------------------------------------------------------

        # The step function takes an action in the Action Space as an argument
        # This then updates the environment and provides a reward
        def step(ActionNumber):

            if a container has arrived to the stack:
                Place the container on a valid stack
                Set reward for move (typically -1)
                Increment move counter

            elif the ActionNumber is anything but "wait":
                Move container from initial stack to target stack
                Set reward for move (typically -1)
                Increment move counter

            else:
                Do nothing
                Set reward for move (for no move, 0)

            Increment the system time by one time step

            #Determine if the simulation has ended
            if system time >= RunTime:
                indicate that the simulation is "done"

            elif there are no legal actions remaining:
                indicate that the simulation is "done"
                Set a reward for this state (typically -50000)

            else:
                indicate that the simulation is not "done"

            return reward and if simulation is "done"
```

Figure 5-9. Pseudocode for attribute initialization and *step* function of the *GymStackEnvironment* class

The *LegalActions* function is another feather of Gym. The *LegalActions* function is used to determine what subset of actions in the action space are available to the environment at a given time step. The purpose of the *LegalActions* function is twofold: first, it forces the agent to select moves that do not violate the dynamics of the system – for example, a container cannot be placed on a full stack; second, it can reduce the size of the action space to improve reinforcement learning training and performance – for example, by forcing a container that is set to depart the system to move to the exit node. The *LegalActions* method initializes two variables – the set of containers that need to move (containers that are arriving or departing), and a list of integers corresponding to the set of legal actions in the action space. The method first checks if any containers are flagged for arrival or departure. If there are arrivals, the method checks for stacks with empty container slots and stores these values to a list. This list is used to generate the valid action pairs which correspond to the list of integer actions in the action space (from the arrival node at element 0 to an open element of the container stacking block, [0,1], [0,2], etc.). Next, the list checks if the container is scheduled to depart and is on top of the stack. If this is true, the legal actions are restricted to one choice – from the current stack to the exit node. Finally, if a container is scheduled to depart but is not on top of its stack, the set of legal actions is restricted to moving containers from the departing container's stack. This reflects the reality of a container stacking yard – if a truck is awaiting a container, the controller will only make moves to retrieve the target container as quickly as possible. If no containers are arriving or departing, the method allows all intra-stacking block moves (reshuffles) or no action. The method returns the set of legal moves as a list of integers corresponding to the Dictionary object in the action space.

```
# The legal actions function provides a list of legal actions
# Legal actions indicate which moves are physically possible
# Legal actions also reduce the size of the action space to improve training
def LegalActions():
    Initialize list of containers that are arriving or departing
    Initialize set of keys for the dictionary of moves in the action space

    if there is a container that is arriving or departing:
        #If a container is arriving
        if a container is arriving to the stack:
            Identify the container stacks with open tiers
            Add keys of current stack/target stack pairs to the list of legal moves

        #If a container is departing
        elif a container is departing the stack and on the top tier of the stack:
            Add key of current stack/exit node pair to the list of legal moves

        else:
            Add keys of current stack/target stack pairs to the list of legal moves

    else:
        Add keys of all valid moves within the stack to the list of legal moves

    return the list of valid moves

# Provide a visualization of the Container Stack Environment
def render():
    Print the simulation time to terminal
    Print the move counter
    Show the Stack Environment

# Provide the shape of the environment for training
def get_observation():
    return a snapshot of the environment for training

def reset():
    Reset the observation state to the t=0 configuration
    Reset the simulation timer
    Reset the move counter
```

Figure 5-10. Pseudocode for the LegalActions, render, and get_observation methods of the *GymStackEnvironment* class

The *render* method is used to provide a visualization of the environment. For this simulation, the render function prints the stacking block to the terminal, displaying a single container parameter at a time, such as arrival time, departure time, or ID. The *get_observation* function returns a snapshot of the environment for training. Occupied slots are represented by a floating point 1.0, while empty slots are represented by a 0.0. The *reset* method is used to return the simulation to the *t=0* state. The function calls the *resetStack* method of the *ContainerStack* class to generate a new environment, resampling the set of containers from the given distributions.

Table 5-5. Overview of the key variables in the *GymStackEnvironment* class

Variable	Data Type	Units	Description
Time	Integer	Minutes	The term used to track the runtime of the simulation. Each time the *step* method is called, the time variable is incremented by one. The simulation ends when this variable is equal to the maxruntime of the ContainerStack object in the Observation Space
Observation Space	ContainerStack	None	The randomly generated ContainerStack object which the agent performs actions on. The Observation Space is manipulated by the *step* method at each time step.
Action Space	Dictionary	None	The set of integer:move pairs representing all possible actions within the environment. At each time step, the agent selects an action to perform in the Observation Space.
Move Counter	Integer	None	A counter representing the number of touches for a given simulation.
needMove	List	None	A list which stores the containers that are either arriving to or departing from the container stacking block. This assists with the development of the set of legal moves at a given time step

Table 5-6. Overview of the key methods in the *GymStackEnvironment* class

Method	Parameters	Returns	Description
step	Integer	Reward, completion flag, metadata	Used to run a single time step within the container stacking environment. The *step* method takes an integer as an argument – the integer corresponds with the set of legal actions provided by the *LegalActions* method. The *step* method also contains the logic necessary to differentiate various types of moves – from the arrival node to the stacking block, intra-block moves, and no action. The method also returns the value of the reward for the step (typically -1 for a container move and 0 for no action), the flag for ending the simulation (if the full runtime has elapsed or there are no legal moves), and metadata from the *get_Observation* method.
LegalActions	None	List	Used to create and return a subset of legal actions from the full action space. There are two types of restrictions: dynamics restrictions and training restrictions. Dynamics restrictions limit the set of legal moves based on the dynamics of the environment. For example, a container cannot be placed on a full stack, cannot enter or leave the stacking block before it is ready, or be returned to the arrival node. Training restrictions are in place to improve the performance of the reinforcement learning algorithm. For example, a container that is flagged for arrival must be placed in the container stacking block before any other actions can occur. If a stack has a container that is scheduled to depart, only containers from this stack may be moved until the target container reaches the exit node.
render	None	None	Used to print the environment to the terminal. Used primarily for assessment of training performance and for testing new features.

get_ Observation	None	Array	Used to provide the format of valid observations. In this case, the observation format is an array in the shape of the Observation Space, indicating which container slots are filled and which are open.
reset	None	None	Used to return the Observation Space to the *t=0* state, reset the environment time variable, and empties the list of containers that need moves. Utilizes the *resetStack* method of the *ContainerStack* class to generate a new random environment sampled from the established distributions. The method can create an entirely new stack or produce a stack of the same original configuration – with the same number of containers in the stacking block at *t=0*, each with the same arrival times, varying only the departure times.

5.4 Simulation performance

This section provides an overview of how the simulation is operated on a small example problem. The purpose of this demonstration is to provide context for understanding the performance of the simulation for the development of the reinforcement learning algorithm. Figure 5-11 describes the code executed to generate the *GymStackEnvironment*. The stack in this example has seven elements (five stacks, an arrival node, and an exit node), five tiers, 10 arriving containers, a maximum runtime of 1200 time steps, and a dwell time parameters of *N(100, 25)*.

```
# Create block with seven elements
# Five tiers
# Runtime of 1200 minutes (~2.5 days)
# Dwell Time N(100, 25)

stack = Stacks(7)
```

Figure 5-11. Code used to create an instance of the *GymStackEnvironment*

Figure 5-12 describes the code used to generate the set of all legal actions and the output of the code. There are 31 total actions in the environment – 30 valid moves and one action with no moves called "pass".

```
#The set of all actions
print(stack.action_space)
{0: [0, 1], 1: [0, 2], 2: [0, 3], 3: [0, 4], 4: [0, 5], 5: [0, 6], 6: [1, 2],
7: [1, 3], 8: [1, 4], 9: [1, 5], 10: [1, 6], 11: [2, 1], 12: [2, 3], 13: [2,
4], 14: [2, 5], 15: [2, 6], 16: [3, 1], 17: [3, 2], 18: [3, 4], 19: [3, 5], 20:
[3, 6], 21: [4, 1], 22: [4, 2], 23: [4, 3], 24: [4, 5], 25: [4, 6], 26: [5, 1],
27: [5, 2], 28: [5, 3], 29: [5, 4], 30: [5, 6], 31: 'pass'}
```

Figure 5-12. Code used to show the entire action space of the *GymStackEnvironment* instance

Figure 5-13 describes the code used to observe the environment at $t=0$, including the arriving containers and initial container stacking block. The code also generates the set of legal actions at $t=0$. As there are no stacks with containers on the maximum tier (five), and no containers scheduled for arrival or departure, all intra-stacking block moves are legal, as well as the "pass" action. The first row of containers is the set of 10 arriving containers. The following five rows are the stacks in the stacking block, where the first element (the value furthest to the left) is the bottom of a stack. The final row, initially empty, is the exit node of the simulation.

```python
#Arrival times of Containers
print("Arrival times of containers")
stack.stack.showStack('arrive')

#Departure Time of Containers
print("Departure times of containers")
stack.stack.showStack('departure')

#Legal Actions
print("Legal Actions")
print(stack.legal_actions())
```

```
Arrival times of containers
[344, 141, 568, 730, 422, 359, 250, 111, 889, 216]
[-68, -5]
[-98]
[-75, 103]
[111, 105, 94]
[127, 9]
[]


Departure times of containers
[449, 253, 723, 794, 511, 461, 353, 219, 987, 308]
[12, 90]
[7]
[55, 193]
[210, 182, 220]
[210, 127]
[]


Legal Actions
[6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 26, 27, 28, 29, 31]
```

Figure 5-13. Code and output of the initial state of the *GymStackEnvironment* instance

Figure 5-14 describes the code used to step through the simulation. The code executes one time step using an action from the set of legal actions, then displays the system time, legal actions, and the stack arrival and departure times. In this case, the action taken is '31' or the "pass" action in which no containers move in the environment.

```
# Input action from Legal actions
# Advance one time step
obs, reward, done = stack.step(31)

print("")
print("Time")
print(stack.t)

print("Legal Actions")
print(stack.legal_actions())


# Arrival
print("")
print("Arrive")
stack.stack.showStack('arrive')

# Departure
print("")
print("Depart")
stack.stack.showStack('departure')
```

Figure 5-14. Code used to execute single time steps and display the updated environment

Figure 5-15 describe the results of taking the first two steps of the simulation, in which both actions taken were ‘31’ or “pass”, with no containers moving. As there are no containers scheduled to arrive or depart, there is no need to take any moves at this time.

```
Time
2
Legal Actions
[6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 26, 27, 28, 29, 31]

Arrive
[344, 141, 568, 730, 422, 359, 250, 111, 889, 216]
[-68, -5]
[-98]
[-75, 103]
[111, 105, 94]
[127, 9]
[]



Depart
[449, 253, 723, 794, 511, 461, 353, 219, 987, 308]
[12, 90]
[7]
[55, 193]
[210, 182, 220]
[210, 127]
[]
```

Figure 5-15. The results of the first two steps of the example simulation

Figure 5-16 shows the results at time steps seven and eight. In time step seven, the container located in the second stack is scheduled to depart the system. This restricts the set of legal actions to only “15”, or the [2, 6] move, removing the container from the stacking block and placing it in the exit node. Once the container has moved, the full set of legal actions is available again.

```
Time
7
Legal Actions
[15]

Arrive
[344, 141, 568, 730, 422, 359, 250, 111, 889, 216]
[-68, -5]
[-98]
[-75, 103]
[111, 105, 94]
[127, 9]
[]


Depart
[449, 253, 723, 794, 511, 461, 353, 219, 987, 308]
[12, 90]
[7]
[55, 193]
[210, 182, 220]
[210, 127]
[]
```

```
Time
8
Legal Actions
[6, 7, 8, 9, 16, 17, 18, 19, 21, 22, 23, 24, 26, 27, 28, 29, 31]

Arrive
[344, 141, 568, 730, 422, 359, 250, 111, 889, 216]
[-68, -5]
[]
[-75, 103]
[111, 105, 94]
[127, 9]
[-98]


Depart
[449, 253, 723, 794, 511, 461, 353, 219, 987, 308]
[12, 90]
[]
[55, 193]
[210, 182, 220]
[210, 127]
[7]
```

Figure 5-16. The output from the terminal at *t=7* and *t=8*, showing the movement of a container from the stacking block to the exit node

Figure 5-17 skips ahead to time step 12, in which a container that needs to exit the stacking block is beneath a container that does not yet need to depart. The set of legal actions is first restricted to the four intra-block moves (from the first stack to all four remaining stacks). In this case, action “6” was chosen. In time step 13, the legal actions are reduced to one action, “10”, moving the container from the stacking block to the exit node.

```
Time
12
Legal Actions
[8, 9, 6, 7]

Arrive
[344, 141, 568, 730, 422, 359, 250, 111, 889, 216]
[-68, -5]
[]
[-75, 103]
[111, 105, 94]
[127, 9]
[-98]


Depart
[449, 253, 723, 794, 511, 461, 353, 219, 987, 308]
[12, 90]
[]
[55, 193]
[210, 182, 220]
[210, 127]
[7]
```

```
Time
13
Legal Actions
[10]

Arrive
[344, 141, 568, 730, 422, 359, 250, 111, 889, 216]
[-68]
[-5]
[-75, 103]
[111, 105, 94]
[127, 9]
[-98]


Depart
[449, 253, 723, 794, 511, 461, 353, 219, 987, 308]
[12]
[90]
[55, 193]
[210, 182, 220]
[210, 127]
[7]
```

Figure 5-17. The output from the terminal at *t=12* and *t=13*, showing the movement of a container within the stacking block

Figure 5-18 skips ahead to time step 111, the first time that a container arrives to the system. The step method is used to execute action “1”, the [0,2] move from the arrival node to the second stack in the environment.

```
Time
111
Legal Actions
[0, 1, 2, 3, 4]

Arrive
[344, 141, 568, 730, 422, 359, 250, 111, 889, 216]
[103]
[]
[]
[111, 105, 94]
[127, 9]
[-98, -68, -75, -5]


Depart
[449, 253, 723, 794, 511, 461, 353, 219, 987, 308]
[193]
[]
[]
[210, 182, 220]
[210, 127]
[7, 12, 55, 90]
```

```
Time
112
Legal Actions
[6, 7, 8, 9, 11, 12, 13, 14, 21, 22, 23, 24, 26, 27, 28, 29, 31]

Arrive
[344, 141, 568, 730, 422, 359, 250, 889, 216]
[103]
[111]
[]
[111, 105, 94]
[127, 9]
[-98, -68, -75, -5]


Depart
[449, 253, 723, 794, 511, 461, 353, 987, 308]
[193]
[219]
[]
[210, 182, 220]
[210, 127]
[7, 12, 55, 90]
```

Figure 5-18. The output from the terminal at *t=111* and *t=112*, showing the movement of a container from the arrival node into the container stacking block

This process is repeated for the 1200-time step duration of the simulation. When the simulation is complete, the reinforcement learning algorithm will call the *reset* function, returning to the *t=0* state and resampling the set of containers. The next following chapter describes the reinforcement learning algorithm, its parameters, and its performance for this study. Further, the next chapter will utilize the algorithm to test the impacts of various changes to the container stacking block environment on performance.

Chapter 6: Integration of reinforcement learning for process control and resilience

6.1 Overview

This chapter describes the integration of the MuZero reinforcement learning algorithm, serving as a controller for the simulation outlined in Chapter 5. The algorithm prioritizes minimizing the number of container touches, an objective of the Port of Virginia, and is able to react to changes in arrivals and departures. The algorithm also provides insights and heuristics which can be included in the existing black box controllers used at the port. Section 6.2 describes a brief overview of the objectives of using the MuZero in the context of this demonstration. Section 6.3 describes the MuZero algorithm in greater detail. Section 6.4 describes the performance of MuZero across several implementations. Section 6.5 outlines several findings derived from executing the MuZero algorithm on the container stack environment. Section 6.6 describes several heuristics discovered by running the algorithm. Section 6.7 describes the application of the MuZero algorithm to the disruptive scenarios as outlined in the previous chapters, exploring how the controller performs under abnormal conditions.

This chapter represents stages three, four, and five of the CPSRM. Both Chapters 5 and 6 cover these stages. However, this chapter focuses on the *Adjust Solutions* decision, as represented by adjusting the parameters of the MuZero algorithm. T2 implements the reinforcement learning solution, while the red team provides feedback to tune results. Figure 6-1 describes the relationship between the content of this chapter and the CPSRM.

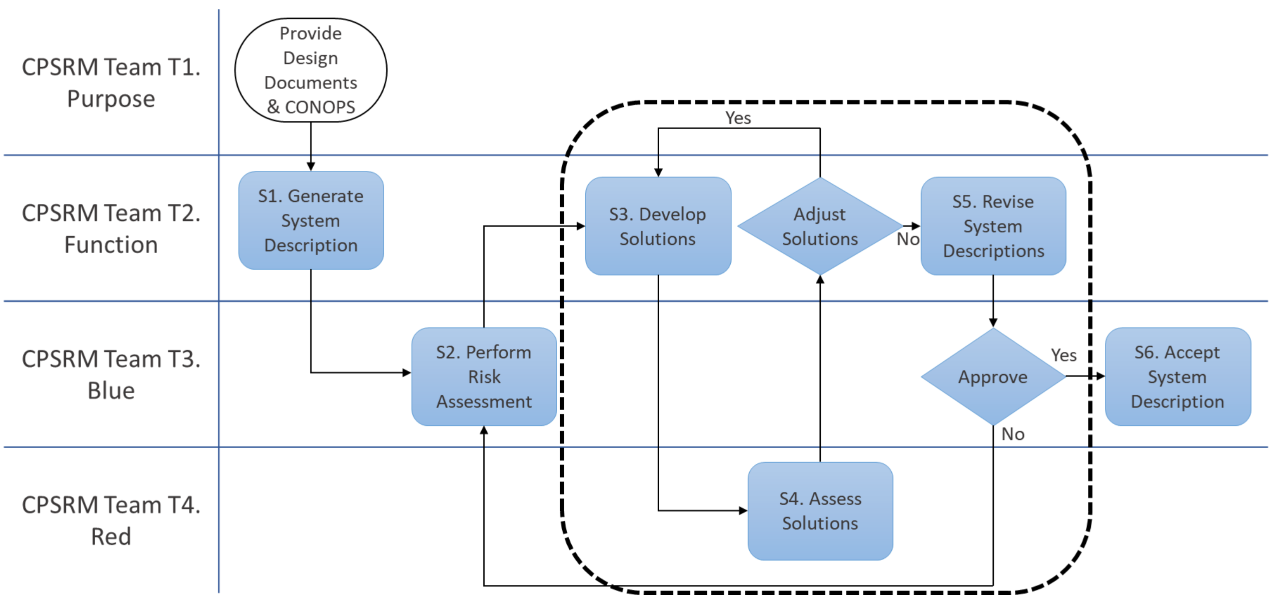

Figure 6-1. Relationship between CPSRM stages three, four, and five, and Chapter 6 – Integration of Reinforcement Learning for Process Control and Resilience. Chapter 6 describes stages three, four, and five of the CPSRM, in which T2 develops a simulation model of the target system to address the operational risks provided from stage two. The red team validates and tests resilience capabilities through reinforcement learning.

6.2 Reinforcement learning for logistics

Ports, including the Port of Virginia, use planning algorithms to improve logistics operations such as for berth allocation and scheduling (Cho *et al.* 2021, Thorisson *et al.* 2019). Such algorithms are useful for improving everyday operations, but also enable the port to be resilient to disruptions. That is, better planning leads to a reduction in the impact of disruptions. While many planning operations utilize machine learning in diction making, port operations are often expert systems governed by heuristics and knowledge (Chou and Fang 2018). This is an effective method for expected operational conditions, but can be strained in times of disruption (for example, the supply chain slowdown of September 2021). To combat this, this dissertation presents a reinforcement learning algorithm approach for controlling container yard operations.

Reinforcement learning is a type of machine learning that trains an agent to take actions (Moerland *et al.* 2023). Reinforcement learning is especially helpful in contexts where traditional optimization is cost or computationally prohibitive. The agent is trained via self-learning, typically in a simulated environment. When the *agent* selects an *action*, it updates the *environment*. The *environment* provides a reward and updates the current *state* of the system. Based on the reward, the agent selects a new action. If previous actions provided positive rewards, the agent is generally more likely to choose these actions again. In this way the algorithm learns the dynamics of the environment and when (in terms of various system states) it is advantageous to take certain actions. The process of choosing certain actions at a certain time is called the *policy*. Developing a new policy is performed during the training phase, when the algorithm explores the impacts of decisions. The agent begins training by choosing random actions and examining the impact to the environment. Actions that yield a positive reward are more likely to be chosen again. However, some decisions are only beneficial given particular state of the environment – the algorithm is able to understand this nuance to make good decisions given particular states. A benefit of reinforcement learning algorithms is that once they have been trained, they can be applied to the real world – either as a fully automated service (robotics) or for providing insight to more complex decisions. Further, these algorithms can be used to test system resilience. There are many hyperparameters to tune the performance and training of reinforcement learning algorithms – these are discussed in greater detail in subsequent sections. Figure 6-2 describes the process of training an agent using reinforcement learning.

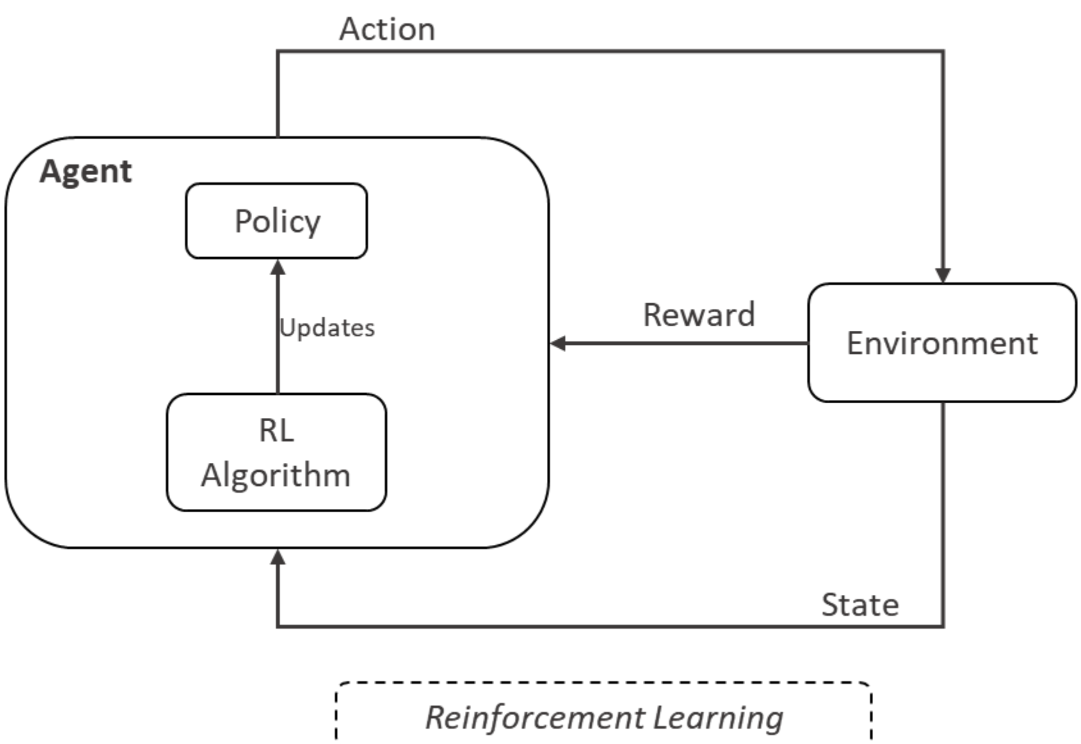

Figure 6-2. Overview of the reinforcement learning process. In reinforcement learning, an algorithm is trained to take actions based on the learned policy. These actions influence the environment, which is updated according to underlying system dynamics. The environment then provides a reward, telling the agent if a given action was “good” or “bad”, and provides an updated state for the next action.

Reinforcement learning is the chosen approach for the development of a control algorithm for several reasons. First, reinforcement learning does not require historical data for training – it learns via self-play on a simulator. Second, reinforcement learning algorithms can handle uncertainty well, especially when compared to other types of machine learning. Third, reinforcement learning algorithms are adaptable to new environments and inputs – an especially useful feature when comparing performance across disrupted and non-disrupted scenarios. Finally, reinforcement learning algorithms are human understandable, providing clear metrics for performance as an output. That is, one can calculate the expected value of a particular decision in terms of points, metrics, scores, or value, a powerful tool for stakeholders. This also enables decision makers to derive heuristics from the algorithm by observing self-simulated environments. The reward for the algorithm is -1 for a container move and 0 for no move, per Port of Virginia specification. The specific reinforcement learning algorithm used in this dissertation is the MuZero algorithm.

The MuZero algorithm and program was developed by DeepMind as a framework for operating in various games or game-like systems. MuZero was first used to achieve superhuman performance in board games such as chess, shogi, and go. MuZero was selected as the reinforcement learning algorithm over other leading algorithms for a few reasons. First, the simulation of the container handling process benefits from the lookahead feature – this allows the algorithm to estimate the arrival and departure times of containers. Other algorithms such as Q-learning are reactive to the state the agent receives (Long *et al.* 2023). MuZero is able to develop more robust strategies due to the lookahead feature. Further, MuZero is a model-based reinforcement learning algorithm – that is, it learns on a model of the environment (in this case a deep neural network) rather than directly from the environment itself. The relative simplicity of the simulation in this dissertation makes it a candidate for the faster and more powerful model-based algorithm than a model free algorithm such as Deep Q-Networks (DQN) (Oroojlooyjadid *et al.* 2023). The decision to use MuZero is also due in part to its ability to scale to larger environments, incorporating multiple variables, objects, and other confounding factors. MuZero also utilizes value (the action taken in a given environment state) and policy (the reward of taking that action) learning networks, contrasting it with other models such as DQN, which only learns the value of a specific action in a specific state. Other reinforcement learning algorithms such as Dyna-Q also take this approach – however, MuZero stands out as it is trained end-to-end (the value and policy networks are trained simultaneously) allowing it to better generalize to new situations and environments (Liu *et al.* 2022).

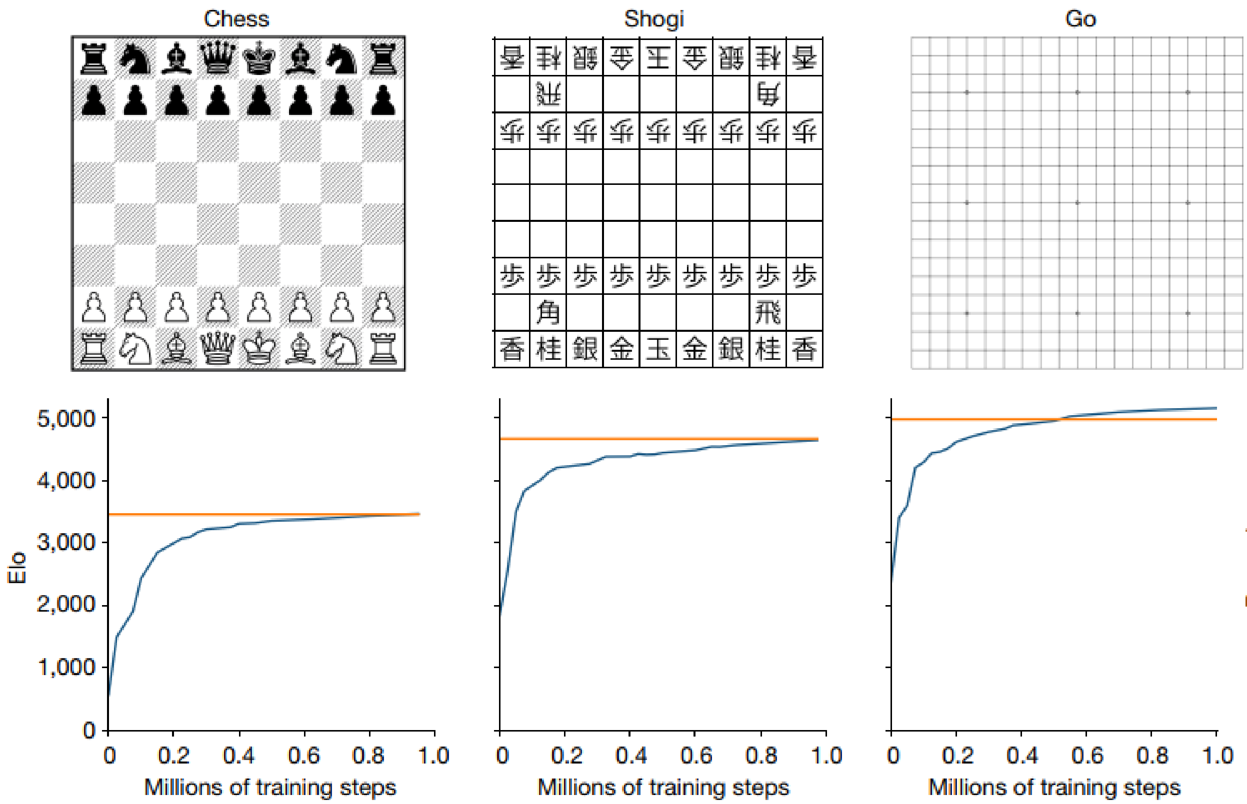

Figure 6-3. Example of the performance of the MuZero algorithm over 1 million training steps across chess, shogi, and go. The orange line represents the highest achieved performance of existing algorithms in terms of Elo. The blue line is the improving performance of MuZero during training. MuZero met or exceeded the best computer performance in these three games. Adapted from (Schrittwieser *et al.* 2020).

In this dissertation, the MuZero algorithm is used to control the actions of the rail mounted gantry crane that manipulates a container stacking block, as simulated using the Gym environment outlined in Chapter 5. The agent develops a policy that controls which actions to take from the action set at a given time step system state. This action updates the environment, which provides a reward and new environment state. This process is repeated across millions of training steps to develop a consistent policy. The reward used for the algorithm is the number of container moves, also called container touches. Each time the agent takes an action moving a container (from the arrival node to a stack, a reshuffle within the stack, or a move from the stack to the exit node), a reward of -1 is provided. If the agent idles, choosing to not move any containers at a given time step, it receives a reward of 0. The objective of MuZero is to maximize rewards, so the agent is incentivized to make as few moves as possible.

Ultimately, the MuZero algorithm is learning the *dwell times* of the containers, discovering where to place incoming containers and how to reshuffle containers currently in the stacking block based on estimates for when the containers depart. This type of uncertainty is where reinforcement learning, particularly MuZero, excels. Subsequent sections outline the technical elements of the MuZero algorithm, the hyperparameters used, and a sample of results.

6.3 Review of MuZero operations

This section outlines the training process for the MuZero algorithm. MuZero receives an observation (for example, the arrangement of containers in the stacking block) as an input and generates a hidden state which is updated iteratively via a recurrent process. This process receives the previous state and an input and outlines a hypothetical next action. Across each of these steps, the algorithm creates a policy p (the next move), a value function v (a prediction of the cumulative reward of the policy), and immediate reward prediction r (the value of playing the move). The objective of the model is to estimate these values, compare the estimates to the realized values, and minimize the difference. There is no requirement that the hidden state capture all information necessary to recreate the original observation. Hidden states can represent any state that correctly estimates the policy, value function, and reward predictions. That is, the agent can perform any physically permissible action that leads to accurate planning (Schrittwieser *et al.* 2020).

Figure 6-4 outlines the planning, action, and training functions of the algorithm. Consider the planning process in **a**. The process begins with an input from a previous hidden state – in the context of recurrent neural networks, a hidden state is the set of inputs used in the current set of operations. Given a previous hidden state s^{k-1} and some potential action a^k (in this case, the set of legal actions at time t), the dynamics function g produces a reward r^k and a new hidden state s^k. The algorithm then uses the prediction function f to generate the policy and value function for s^k, p^k and v^k respectively. s^0 is generated by past observations of the system via the representation function h.

In **b** a Monte Carlo tree search (MCTS) is performed for each timestep t in process **a**. A hypothetical next action a_{t+1} is sampled from the search policy π_t which is based on the visit count for each possible action from the root node. Using this new action, the environment generates an observation o_{t+1} and reward u_{t+1}. These values are stored in the replay buffer, and are used to compare the relative values of one action to another.

In **c**, the algorithm trains the model based on trajectory data stored in the replay buffer. A past observance from $o_1,\ldots,o_t$ is taken as an input. The model then assesses performance recurrently for K steps (until reaching a termination condition). At each k, the dynamics function g receives the hidden state s^{k-1} form the previous step and the action a_{t+k}. At each step the policy, value function, and reward are trained together, end-to-end via backpropagation. Each function is estimated via recurrent neural networks or by traversing value networks. The first objective is to minimize the error between actions predicted by the policy p_t^k and the search policy π_{t+k}. The second objective is to minimize the error between the value function and value target. The third objective is to minimize the difference between the predicted an observed immediate reward.

These are combined into a single loss function. An L2 regularization term scaled by the constant c is added to the loss function. This is outlined in the loss function in Equation 6.1.

$$l_t(\theta) = \sum_{k=0}^{K} l^{\mathrm{p}}(\pi_{t+k}, p_t^k) + \sum_{k=0}^{K} l^{\mathrm{v}}(z_{t+k}, v_t^k) + \sum_{k=1}^{K} l^{\mathrm{r}}(u_{t+k}, r_t^k) + c\|\theta\|^2, \qquad (6.1)$$

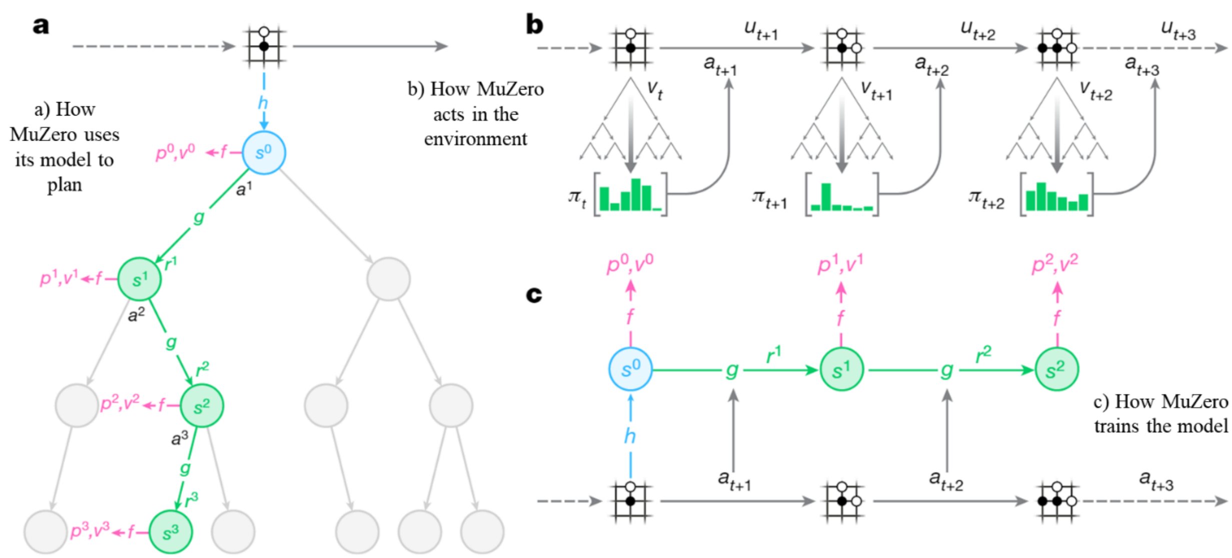

Figure 6-4. An overview of the main processes for training the MuZero algorithm. There are three primary phases – a) the algorithm traverses various system states based on previous states, policy, and values, b) the algorithm performs a Monte Carlo tree search for each timestep in a, and c) the algorithm samples a trajectory of events from the replay buffer and calculates the policy, value function, and reward values for each state. Adapted from (Schrittwieser *et al.* 2020).

6.4 MuZero parameters and performance

There are eight primary structures, parameters, and hyperparameters that make up the MuZero algorithm. These govern how deeply the algorithm examines the MCTS, how the algorithm balances exploration and exploitation of actions, how it forms the underlying dynamics model for the environment, how it optimizes the reward function, and other factors. Table 6-1 describes these elements at a high level and will be discussed in greater detail.

Table 6-1. Overview of the primary parameters and hyperparameters used to train the MuZero algorithm

Parameter/ Hyperparameter	Description
Neural Network	The architecture of the deep neural networks used in the MuZero algorithm. The neural networks are used to derive values for the dynamics model, value function, and policy.
Simulations	The number of future time steps the algorithm utilizes to develop optimal control policies – this is the number of future moves self-simulated by the algorithm when determining which action to take.
Training steps	A training step is a moment in the simulation time in which the MuZero algorithm performs the number of self-play simulations. To master chess and shogi, the algorithm took roughly 1 million training steps.
Temperature	The value used to determine the exploration-exploitation tradeoff when generating a policy. A higher temperature will encourage exploring new actions, while a lower temperature leads to a policy of selecting the current “best” action.
Replay Buffer	The data structure used in many reinforcement learning algorithms (MuZero, DQN) to store prior observations and results of the environment, actions, and rewards. The size of the buffer describes how many prior observations are used to select new actions.
Discount Factor	The weight of future rewards – the higher the weight, the more value ascribed to long-term rewards. Smaller weights give more value to immediate or short-term rewards. The discount factor ranges from 0 (the immediate reward fully dominates the decision) to 1 (all rewards have an equal weight across the time horizon).
Optimizer	The optimizer determines how the parameters of the neural network are updated during training based on the gradient of the loss function. SGD and Adam are two common optimizers used to find solutions.
Learning Rate	The value used in conjunction with the optimizer to determine the step size of weight changes during optimization.

The neural network architecture is a core element of the MuZero algorithm. It is used to approximate the value and policy functions of the model, as well as maintain the dynamics model of the environment. It consists of four neural networks – a value function, policy network, dynamics function, and representation network. The representation network creates the latent representation of the current state, converting the environment into a model – in this dissertation, a fully connected neural network is used. The value function takes the latent representation output (*h)* and generates an estimation of the expected long-term reward from that state – for this dissertation, this is the expected number of moves in a simulation. The policy network takes the latent representation output and provides a probability distribution across all possible actions – that is, each action has an associated probability of selection given a current state of the environment. The dynamics function takes h_t as an input, and generates the next state h_{t+1}. The parameters of the neural network are updated across training steps – the dynamics function is used to simulate future possible scenarios, and these predictions are used to update the parameters of the neural network architecture (Schrittwieser *et al.* 2020).

The number of simulations is a parameter used to determine how many future time steps the algorithm uses to make decisions. At each time step in the simulation MuZero performs a MCTS, selecting actions and generating hypothetical future scenarios. A higher number of simulations allows the algorithm to make more informed decisions by generating a more robust simulation. A lower number of simulations will provide less robust results, but can generate results more quickly (Cutler *et al.* 2014). In this scenario, the number of simulations has a secondary effect – as the departure times of containers are hard-coded (though randomly sampled), the agent has the ability to see when a container will depart during lookahead. This emulates the real-world process of truck scheduling times for picking up containers from the terminal. The information is incomplete – the agent does not look to the end of the simulation, and it still must plan given the uncertainty of departure times.

Training steps are related to the simulation parameter and are a critical element for training the MuZero algorithm. A training step is a moment in the simulation in which the agent takes the observation from the time t, selects an action based on the existing policy, and updates the policy using observations from the replay buffer. Each training step looks ahead based on the number of simulations parameter above – that is, at each training step the algorithm will look ahead in the MCTS to t + *num simulations.* Generally, the more training steps used when training the model, the better the performance of the algorithm as it has more time to learn. However, selecting a number of training steps that is too large can lead to overfitting, reducing the flexibility of the model. To combat this, many RL algorithms, including MuZero, recommend reducing the temperature parameter as the number of training steps rises (Bellow *et al.* 2016).

The temperature parameter (τ) is used to control the exploration-exploitation trade off - that is, the percent of the time that the algorithm chooses the "best" action or selects a new action. The temperature parameter takes a value greater than 0, and typically [0,1]. The visit count of each child node is raised to the $1/\tau$ prior to the selection probability being calculated – this encourages the selection of actions that have not been taken as often. A larger temperature parameter results in greater variety in action selection, while a lower temperature parameter converges on the actions with high visit counts and selection probability. For environments such as the container stacking block with a large number of actions, a larger temperature parameter is preferred (He *et al.* 2018).

The replay buffer is a data structure used to store previous training examples and experiences. The replay buffer stores an environment, action, reward, and updated state of the environment. The results of a potential action are compared to the current action selection at a training step to determine the next best action using previous experience. This helps the algorithm avoid overfitting. The replay buffer is also used to update the parameters of the neural network architecture during training. The replay buffer also dictates the number of “unroll steps” – the number of simulated moves saved from each batch element – and the number of future moves to consider when calculating new weights. The primary tradeoffs on the size of the replay buffer are memory considerations (Zhang and Sutton 2017).

The discount factor is used to determine the relative importance of near-term and long-term rewards. For example, the simulation in this dissertation considers all container moves with equal weight, regardless of simulation time. However, in chess, a stronger move in the short term may be more valuable than strong moves several turns from now. The discount factor is represented by γ and has a value between 0 and 1. When the agent calculates the future rewards of an action, it calculates using the sequence $R = r_1 + \gamma\, r_2 + \gamma^2\, r_3 + \ldots + \gamma^{(n-1)}\, r_n$ where R is the total reward over time steps n, with r_i being the reward received at time i. It follows that a discount factor closer to 1 will give more (or equal) weight to future rewards, while smaller values will reduce the value of future rewards. For the simulation of the container stacking block in this dissertation, the discount factor is 1.

The optimizer is an algorithm used to update the weights and parameters of the neural network. There are many types of optimizers – however, MuZero uses either Adam or stochastic gradient descent (SGD) most commonly. SGD is the simpler algorithm, with less stringent memory requirements and fast learning. It is less likely to overfit than Adam. Adam is more flexible than SGD and adjusts learning rates adaptively. Adam can take larger batch sizes and run more efficiently than SGD, and is better suited for deep networks. The optimizer also takes weight decay and momentum as inputs. Weight decay is a regularization technique that penalizes large positive or negative loss values. Momentum is a tool used in optimizers to assist in faster convergence.

The learning rate of the optimizer determines how quickly the weights update during optimization. A high learning rate can lead to fast convergence, quickly finding an optimal solution. However, the solution may be unstable or overfit. A lower learning rate can take longer to find optimal solutions but has higher degree of accuracy, stability, and flexibility.

6.5 Findings

This section describes the performance and findings of the MuZero algorithm applied to the container stacking block simulation. The simulation presents a reduced-size environment for testing the algorithm, decreasing training times and computational resource requirements. The model is run using research computing resources with 24 cores, 192 GB of memory, and four GPUs with 12 GB VRAM each. The model was trained using the parameters outlined in Table 6-2. It takes roughly 12 hours for the model to converge on a solution for an environment with 10 stacks, five tiers, and ~175 containers entering and exiting the system. This contrasts with an 8 lane x 30 bay x 5 tier stacking block at the Port of Virginia, with roughly 20 containers/hour arriving to the block continuously during operating hours. The algorithm is executed from a command line and written in the Python programming language, leveraging the PyTorch machine learning framework (Duvaud and Hainaut 2020).

To execute the program, the user runs the module in a command line terminal and selects the simulation of interest. Figure 6-5 describes the initial output to the terminal. *0: Train* is used to execute the algorithm and begin training the model. *1: Load pretrained model* is used to load a previously trained model for further training, analysis, or other uses. *2: Diagnose model* is used to access various metadata about a trained model such as information about the trajectory (rewards, policies) at each time step, information about the optimizer, and visualizations of the MCTS. *3: Render some self-play games* provides a visualization of a random, simulated environment and executes the simulation using the loaded model. *4: Play against MuZero* is used for multiplayer environments and allows a human to compete with the trained models – this feature is not available for the container stacking block simulation. *5: Test the game manually* allows a user to select actions in the simulated environment. MuZero will recommend a move, but the user is free to take any legal action – MuZero can then score the quality of the move in terms of policy and reward. *6: Hyperparameter search* reveals the hyperparameters used in the currently loaded model. *7: Exit* terminates the program, unloading the model from memory.

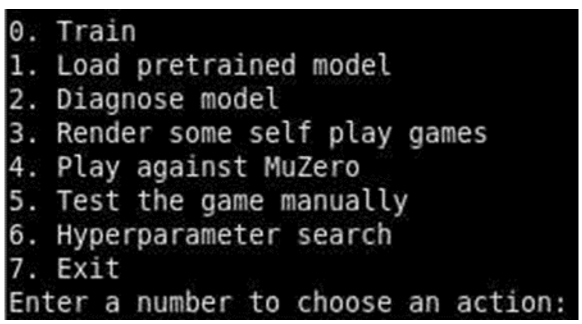

Figure 6-5. The main selection screen for the implementation of the MuZero algorithm. There are seven actions available.

Training the MuZero model is computationally expensive for conventional desktop computers. Figure 6-6 describes the memory usage for a typical run of the algorithm on the container stack environment with 10 stacks, five tiers, and ~150 incoming containers over a 1200-time step simulation. The model takes roughly 44 GB of memory to operate and train. For comparison, the model for the full Port of Virginia container stacking block with 240 stacks has a nearly 150 GB size requirement and a nearly constant 92% CPU usage rate. While training, the algorithm provides a summary of information to the user.

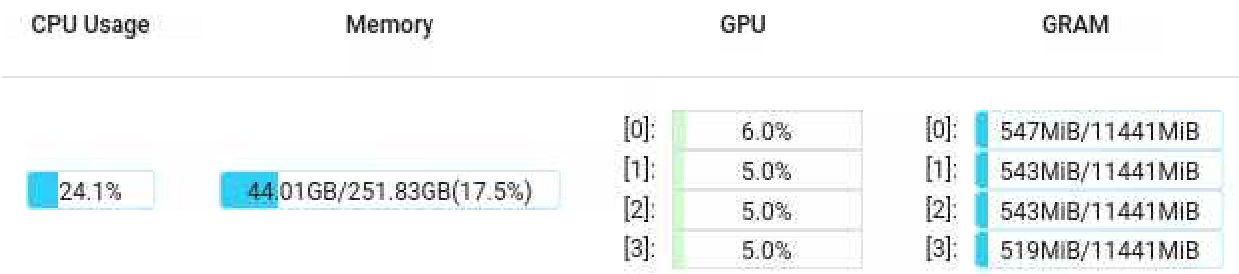

Figure 6-6. Example of the computational requirements to run MuZero on the simulated container stacking block environment.

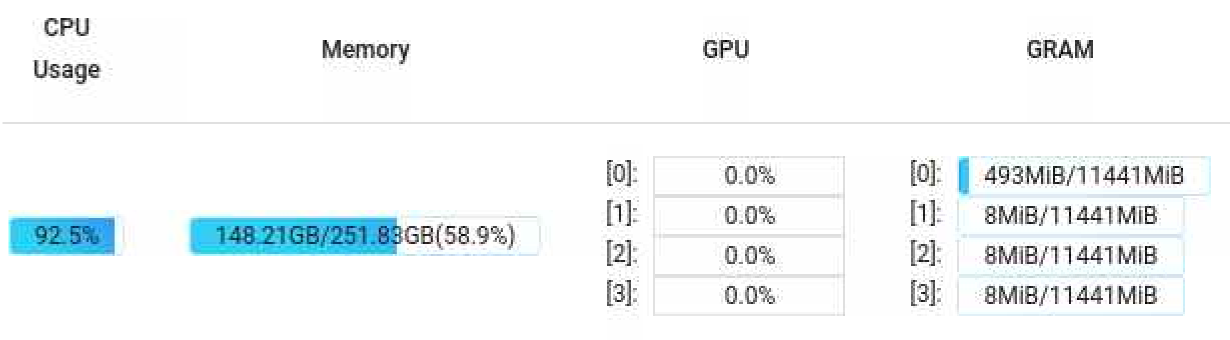

Figure 6-7. Computational requirements for a full Port of Virginia container stacking block (30 bays, eight lanes, five tiers, for 1200 total container slots)

Figure 6-8 describes the summary information provided by the algorithm during training. The *Last test reward* describes the total reward from the last completed game – in this case, the last completed simulation had a total of 760 moves – roughly 4.4 touches per container. The *Training step* shows the current training step of the training cycle. *Played games* shows the number of completed games. *Loss* shows the value of the loss function at the last training step.

```
Last test reward: -7600.00. Training step: 81227/100000. Played games: 80. Loss: 143.44
```

Figure 6-8. Summary information provided in the command line terminal during training.

While this information is useful, it only provides a snapshot of performance. The software produces several graphs showing how performance changes over training steps. Figure 6-9 describes the policy, reward, value, and total weighted loss values for an implementation of MuZero. This model, whose parameters and hyperparameters are described in greater detail later in this section, began to converge on a solution around training step 30,000. It converged on an optimal policy around training step 10,000. It also converged on a on an optimal reward loss around training step 10,000. This indicated that the algorithm was quickly able to develop short term policies. The value loss function, the function responsible for calculating and estimating the total reward for a given policy, converged around training step 30,000. The total weighted policy loss, the weighted sum of the three other loss functions, begins to converge at step 10,000, and reaches stability around step 30,000.

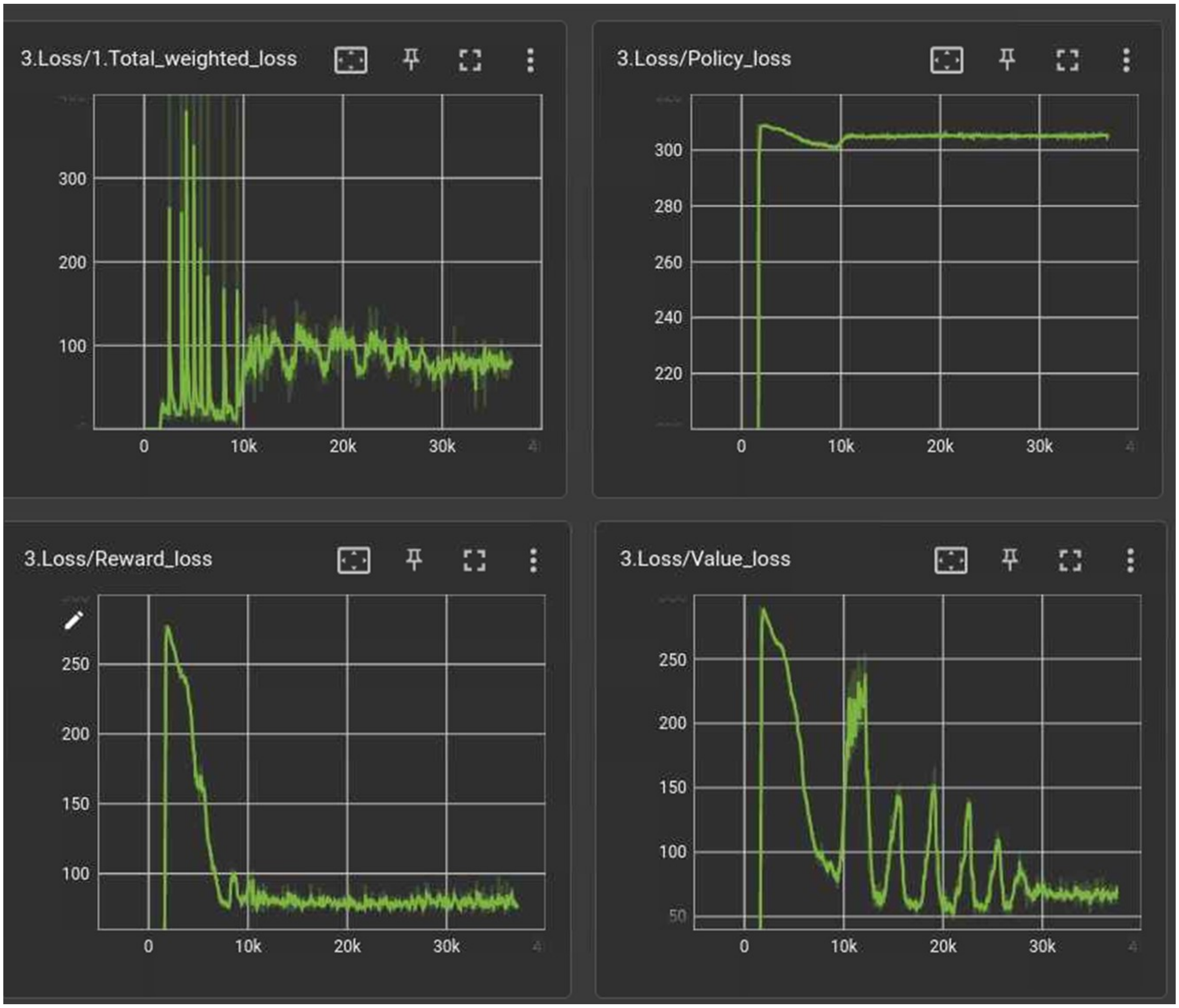

Figure 6-9. Graphs depicting the policy, reward, value, and total weighted loss values over roughly 35,000 training steps for an implementation of the MuZero algorithm

Table 6-2 describes the parameters, hyperparameters and architectures used for the model trained in Figure 6-9. These parameters were used to generate the experimental results outlined below. These values were derived over dozens of experiments via trial and error. The model balances complexity, training time, and performance. Table 6-3 describes the relevant simulation environment parameters used to develop a baseline performance

Table 6-2. The set of parameters, hyperparameters, and neural network architectures used to develop results for the MuZero algorithm applied to the container stacking problem.

Parameter/ Hyperparameter	**Description**
Neural Network	Fully connected network: • Support size: 40 • Encoding size: 25 • Representation layers: 1 • Dynamics layers: 16 • Reward layers: 16 • Value layers:16 • Policy Layers: 16
Simulations	240
Training steps	100,000
Temperature	0.25
Replay Buffer	Replay Buffer: • Buffer size: 100 • Unroll steps: 64 • Future steps: 64
Discount Factor	1
Optimizer	SGD • Weight decay: 0.0004 • Momentum: 0.9
Learning Rate	0.005, constant learning rate

Table 6-3. Overview of relevant environmental parameters used to measure the performance of the MuZero algorithm. This framework is used in the baseline algorithm used for analysis.

Environment	Value	Description
Number of Arrivals	150	The number of containers arriving to the simulation during the simulation time. All containers are generated per the logic outlined in Chapter 5, with a randomly sampled arrival and dwell times.
Simulation Time	1200	The length of the simulation in discrete time units. As the real-world terminal takes an average of three minutes per container move, three times this value is the number of minutes of simulation. The 1200-time units for the example case is equal to roughly 7.5 days of operation.
Dwell Time	$N(200, 50)$	The dwell time distribution from which the randomly generated containers are sampled. In this example case, the dwell time is roughly equal to 1.25 days of waiting in the container stacking block.
Stacks	10	The number of container stacks used in the simulation for the example case.
Tiers	5	The number of tiers in the container stacks in the baseline example case

A primary objective of applying MuZero to port operations is to reduce the number of touches per container. As the MuZero algorithm completes simulations and examines tens of thousands of possible future conditions, there is no single value for touches per container for a given model and environment – however, an average range of touches per container is provided. The algorithm was applied to a container block with 10 stacks, five tiers, and 150 arriving containers over 1200 training steps. The stacking block is prepopulated with a random sample of containers at roughly 50% utilization at the start of the simulation.

A benefit of using these types of models, especially in the context of designing resilience for CPS, is the ability to compare the relative performance of different environmental configurations. This can provide critical information to the port regarding new stacking policies. Results are tested across four models. There is a test model (greatly reduced arriving containers), the baseline model as described in previous sections, a "tall" model trained on an environment with six tiers rather than five, and a "wide" model trained on an environment with 12 stacks rather than 10. Table 6-4 describes the results for the four models. The two metrics of interest are the average utilization of the stacking block and average touches per container. Utilization refers to the percentage of container slots that are occupied at a given time. Utilization can vary greatly depending on several factors – utilization is the average occupancy rate of the container stacking block during the period of the simulation in which arrivals are added to the stack. Touches per container also varies depending on several factors – early and late in the simulation, when utilization is lower, there are substantially fewer touches per container. A range of touches per container are provided to consider these factors.

The test scenario had the best performance. Given the relatively few arrivals to the environment, utilization remained low and drove down the total number of touches. The model was able to take no action during most time steps. The optimal number of touches per container is two (one arriving and one departing), and the performance of 3.5 to 3.7 touches per container is considered good performance – consider that the Port of Virginia currently realizes between 4 and 4.5 touches per container. The baseline model ranges from 4.6 to 5.6 touches per container, with an average utilization around 80%.

Compare these results with the "tall" model in which each of the 10 stacks has the maximum number of tiers raised to 6. There are now 60 container slots rather than 50 as in the base model. The number of arrivals is held constant. The tall model sees slightly improved performance from the baseline model, with a range of 4.6 touches in favorable conditions (lower utilization) to 5.3 touches in unfavorable conditions (higher utilization). The tall model realizes an average of 75% utilization. The tall model has no improvement over the baseline model in favorable conditions, with both models at about 4.6 touches per container. There is an improvement in performance during unfavorable conditions, from 5.6 touches in the baseline to 5.3 in the tall model. The tall model demonstrates the same heuristics as the baseline model, favoring one or more empty stacks when possible, and broadly keeps stacks below their maximum height. Heuristics are explored in greater detail in subsequent sections.

These results are compared to the "wide" configuration model, in which there are two additional stacks for a total of 12, each with five tiers – again with 60 container slots as opposed to the 50 in the baseline. The wide model had better performance than the baseline and tall models in both favorable and unfavorable conditions, with 4.3 touches per container in favorable conditions (0.3 fewer than the baseline and tall models), and 5.2 touches per container in unfavorable conditions (0.4 fewer than the baseline and 0.1 fewer than the tall model). A notable difference in performance is the number of time steps in which the wide model makes no container moves. Through the first 80% of the simulation, the most active portion of the simulation, the baseline and tall models make no moves in 2-5% of time steps. Contrast this with the wide model, which takes no action in 8-10% of available time steps. That is, the additional stacks are more valuable in terms of touches per container than additional tiers. This information can be used to further refine the system description for the CPSRM

Table 6-4. Overview of specialized model performance trained on augmented environments

Parameters					**Results**	
Model	**Sim Time**	**Stacks**	**Tiers**	**Arrivals**	**Utilization**	**Touches/container**
Test	1200	10	5	**25 (~50 total)**	~30%	~3.5 - 3.7
Baseline	1200	10	5	150 (~175 total)	~80%	~4.6 – 5.6
Tall	1200	10	**6**	145 (~175 total)	~75%	~4.6 – 5.3
Wide	1200	**12**	5	145 (~175 total)	~75%	~4.3 – 5.2
Port	N/A	240	5	20/hour	Varies	~4 - 4.5

6.6 Heuristics

Another useful feature of MuZero and reinforcement learning in general is the ability to examine the agent while it operates in the environment to identify heuristics and patterns that the algorithm discovers. Figure 6-10 describes a visualization of the baseline MuZero environment. There are eight primary elements of the visualization. First, the tree depth is an integer that indicates how many future nodes the algorithm examines to make a decision. Second, the *root value for player 1* (recall this simulation has a single player) indicates the expected reward for the simulation given the current state and lookahead conditions. Next is the played action, which indicates which action was played at the last time step that led to the current state. Simulation time and container moves indicate how many time steps the simulation has taken, and how many container moves have been made to this point. The next element is a list of arrivals – this is the set of containers that are scheduled to arrive to the stacking block during the simulation time. Containers are listed by their ID number, beginning at 0 and incrementing by one for each container in the simulation. The next set of elements (usually elements for this dissertation) is the set of stacks in the container stacking block. The stacks are oriented vertically, from left to right.

That is, each row is a different container stack, and each element in each row is a container. Containers at the beginning of the row – the first element and further left – is the bottom of the stack. Containers to the right are stacked "on top" of containers to the left – that is, if the agent takes an action to move a container from a given stack, it will pick the last element of the row. The final element of the list is the exit node – this is where containers that have departed the simulation are stored. Finally, the user manually moves the simulation forward one time step – the agent selects an action according to policy and updates the environment. For this demonstration, the stacking block and exit node will be the primary elements of interest for deriving performance heuristics – the tree depth, root value, action played, simulation time, container moves, and arrivals are often excluded for clarity of view. The heuristics that the algorithm learns can be applied to new and unknown systems as well.

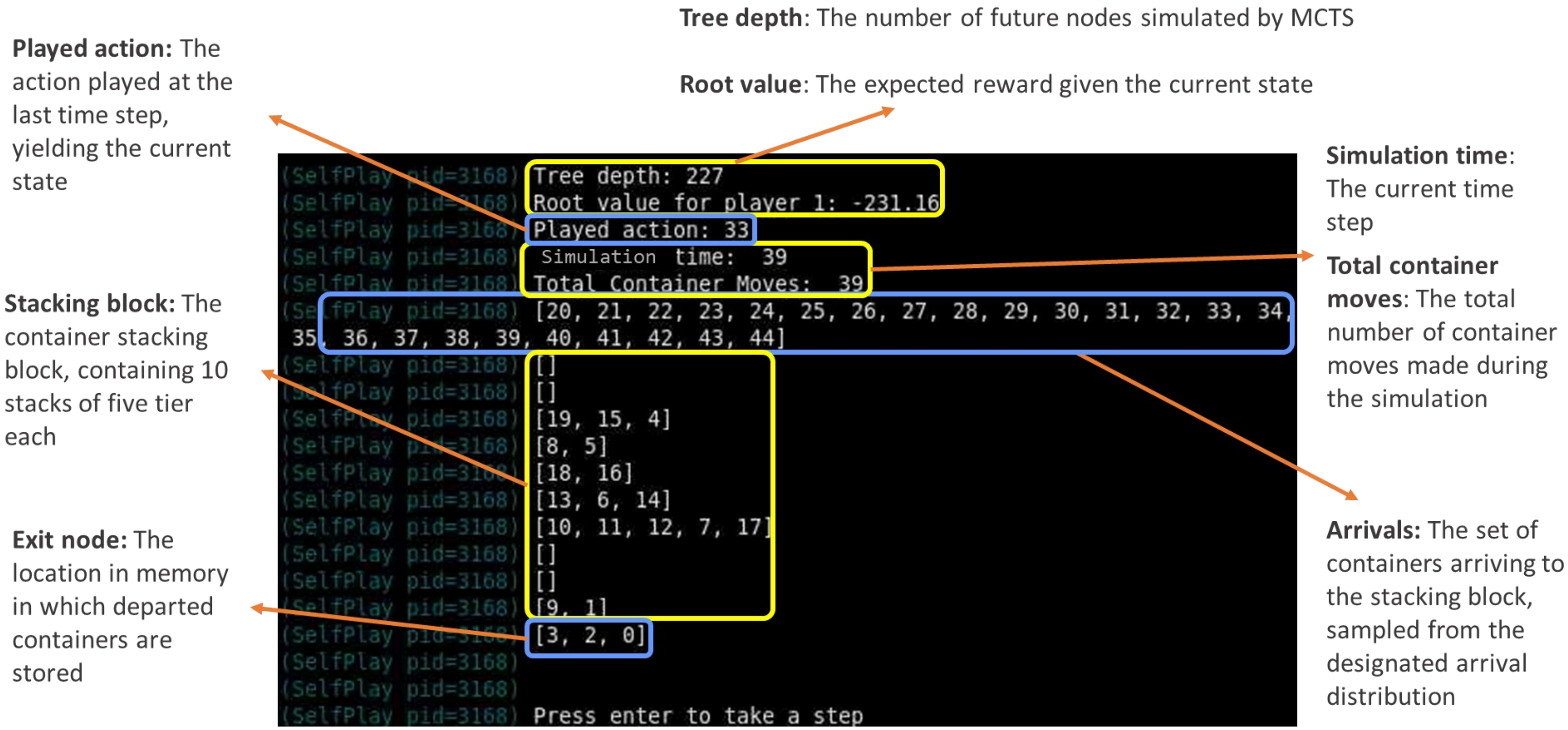

Figure 6-10. Overview of the simulated stacking block environment.

Figure 6-11 describes the beginning stages of a simulation using a trained MuZero model as the agent with the architecture of Table 6-2 and the inputs of Table 6-3. There are 24 containers in the initial stack, with 150 incoming arrivals. At the beginning of the simulation, in time step one, the algorithm moves container 0 from the first to the tenth stack. This highlights a recurring theme and important heuristic discovered by MuZero – the algorithm strongly favors leaving empty stacks. By time step 23, the algorithm has created two empty stacks, and by time step 55 three stacks are empty. This is the case despite eight new containers arriving to the stacking block. There are several potential reasons for this. First, when reshuffling containers to reach a container on a lower tier, the empty stacks are used to temporarily store containers. Second, new arrivals are often placed on an empty stack temporarily before being moved to a higher tier stack.

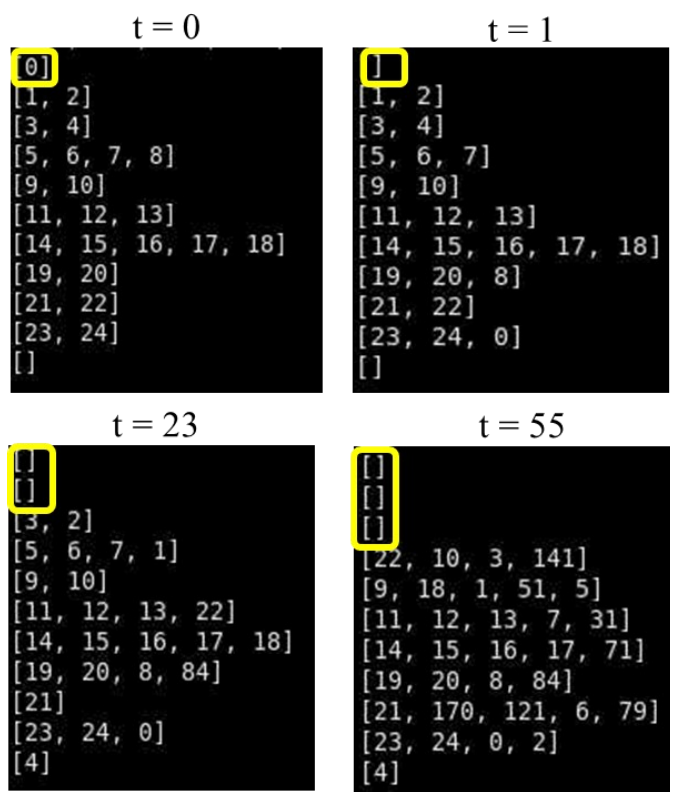

Figure 6-11. Demonstration of the MuZero algorithm prioritizing the creation of empty stacks early in the simulation

Figure 6-12 demonstrates the use of empty stacks to reach lower containers. At time step 61, container 15 has reached its departure time and needs to be moved to the exit terminal. To do this, containers 16 and 17 must be moved first. Container 17 is moved to the empty first stack in time step 62. Container 16 is moved to the top of the fifth stack. Container 15 is then moved to the exit node. By time step 68, containers 16 and 17 have been returned to the seventh stack. This process – reshuffling containers to separate and empty stacks – is seen frequently when observing the performance of the algorithm.

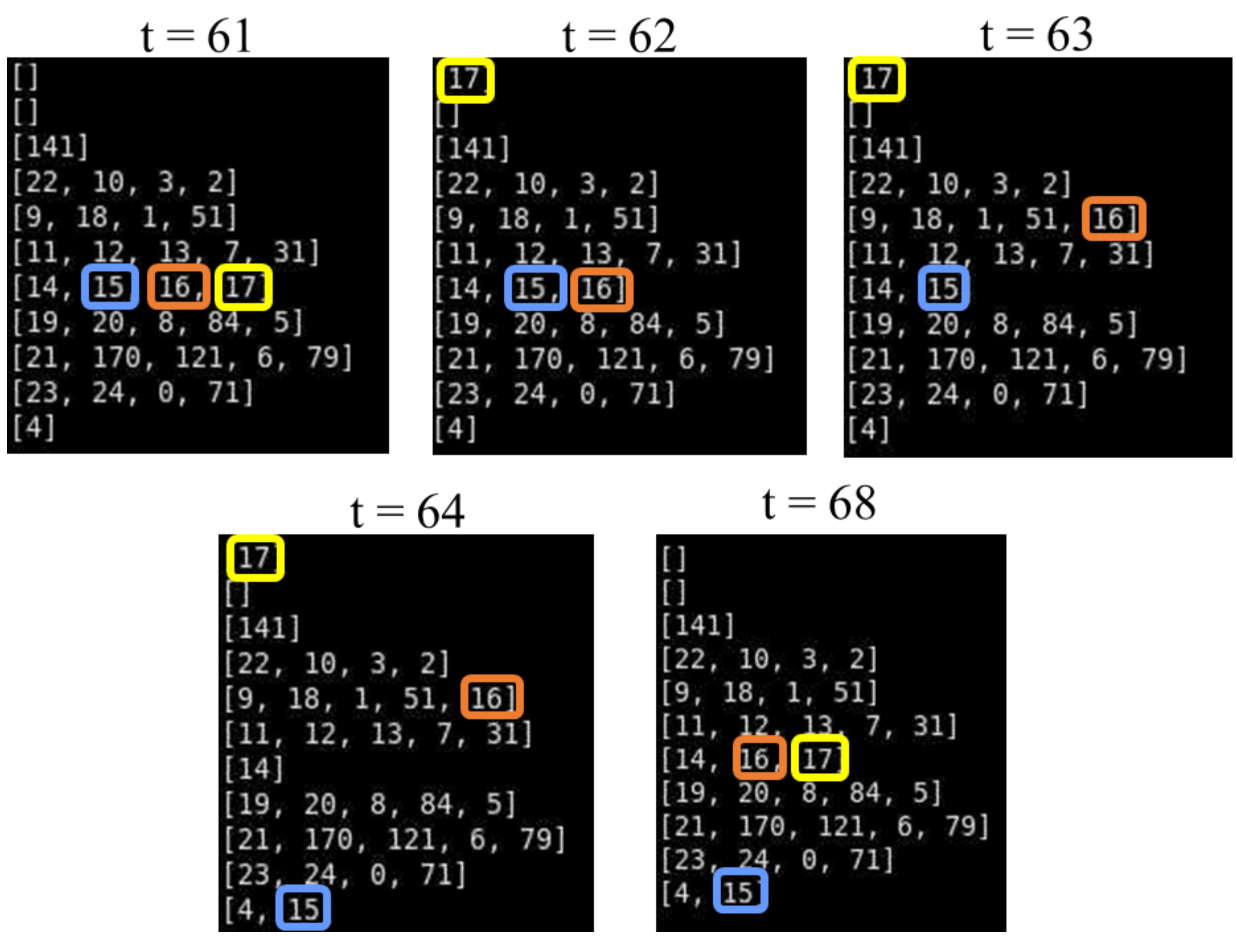

Figure 6-12. Overview of a reshuffle to place a departing container in the exit node. The algorithm utilizes an empty stack as one of the destinations for the reshuffle before returning the container to its starting stack.

The algorithm also uses the empty stacks as storage for new arrivals. New containers are often placed in the empty stacks first before being moved to taller stacks. Figure 6-13 describes this process in which a new arrival, container 116, is placed first in the empty first stack. The agent performs a few reshuffles before moving container 116 to the fifth stack. There are a few potential reasons for this. First, when the stack is at a low utilization, there is a chance that a new container will only be touched twice (initial placement and departure). When the number of touches per container for a given simulation is around 4.5, two touches is a good reward. Second, placing the new container on an empty stack prevents it from blocking a container scheduled to depart sooner. The lag between the initial placement and reshuffle allows the agent time to arrange the stacking block in a more advantageous state. As this stacking block is at 76% utilization, the agent had limited options for placement.

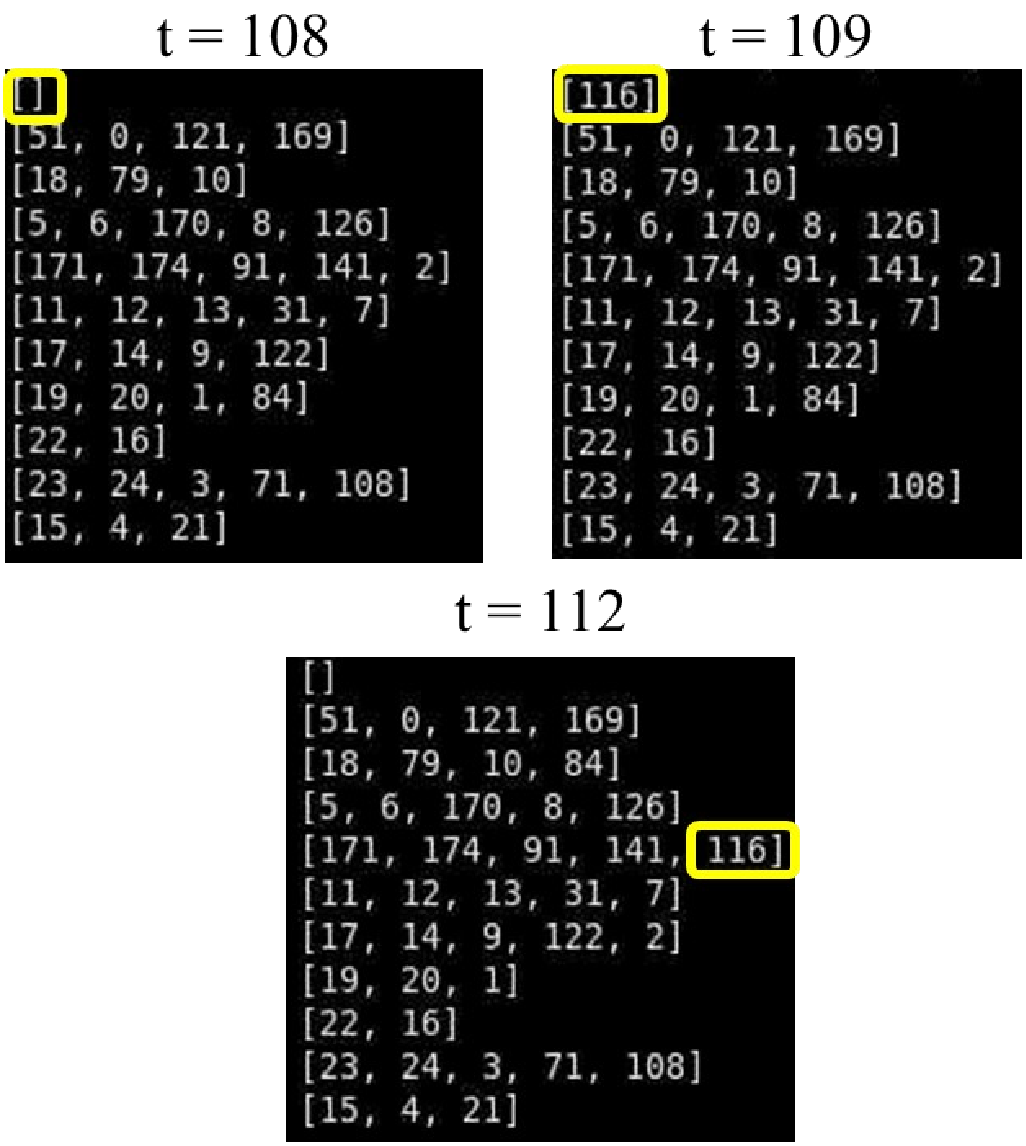

Figure 6-13. Empty stacks are often used as for storing arrivals. These are then moved to taller stacks within a few time steps.

Figure 6-14 provides a demonstration of the algorithm reshuffling over several time steps to enter a more advantageous position. Broadly, containers in the initial stacking block (containers 0 through 24 in this simulation) will exit the stacks before new arrivals (25 through 174). In time step 224, several initial containers are at the bottom of stacks – containers 9, 18, and 0 are all below new arrivals. By time step 243, these initial containers are at the top of their own stacks, ready for departure. Further, note the utilization of the empty second and third stacks for the reshuffling operation. This is not a fixed rule – note that container 7 remains beneath four new arrivals, and containers 3 and 22 are below two new arrivals. However, there is a tendency for the algorithm to group containers from the initial stack together, separated from the new arrivals.

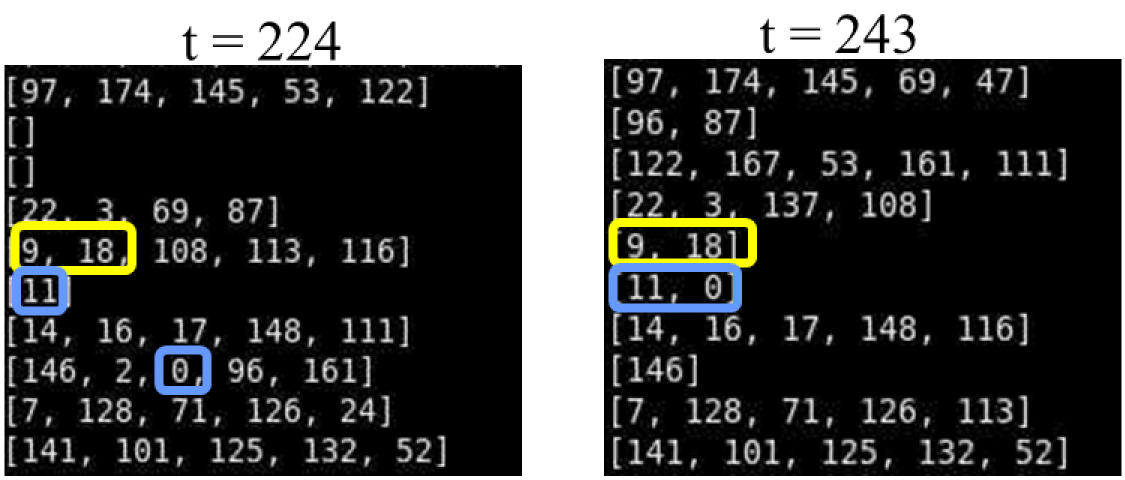

Figure 6-14. Demonstration of the algorithm self-correcting out of a difficult system state

Another heuristic is the tendency of the algorithm to ignore the highest tier container slot in a stack. That is, if there are five tiers, the algorithm will attempt to fill as many stacks to four tiers as possible before moving to five. Figure 6-15 demonstrates this heuristic, as four of the ten stacks are filled to four tiers at time step 271. However, this heuristic is secondary to the creation of empty stacks – consider the same simulation at time step 400. Six of the ten stacks are full, while two are completely empty.

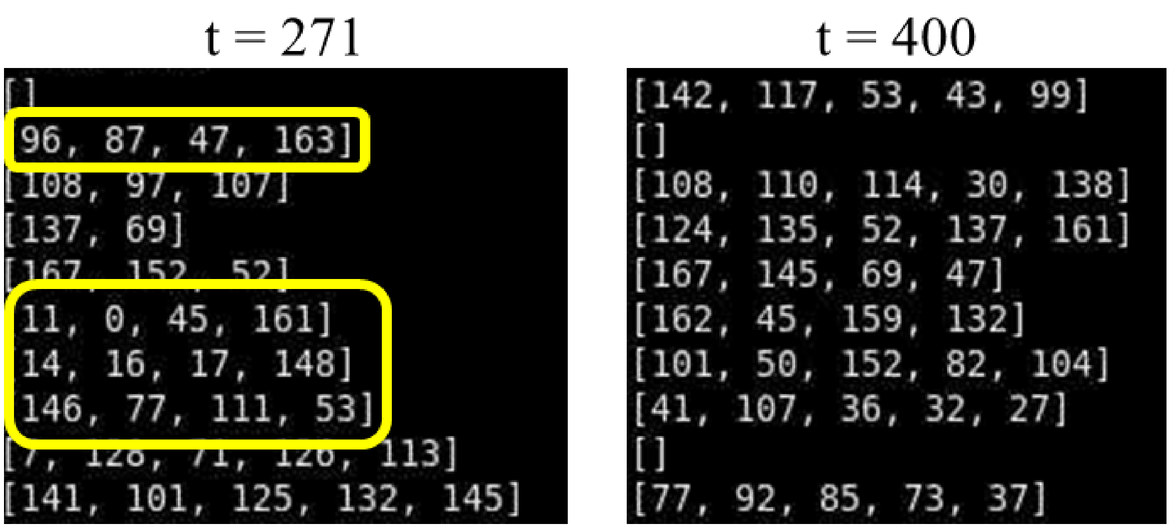

Figure 6-15. Demonstration of the algorithm filling stacks to one fewer tier than the maximum value. This heuristic is not favored over the empty stack heuristic.

It should be noted that there is an option for the algorithm to make no moves and take no action. Figure 6-16 describes this simulation at time step 1146. At this time, there were 973 total container moves. This means that the agent chose the “no move” action 173 times. This indicates that the algorithm learned to take no action, when possible, as there are still containers in the stacking block at this time step. However, many of these no move actions were taken toward the end of the simulation when stack utilization was very low (at time step 1146, it is at 12%). Ultimately, the simulation took 979 total moves to place 174 containers for a final touches per container of 5.6 over the entire run.

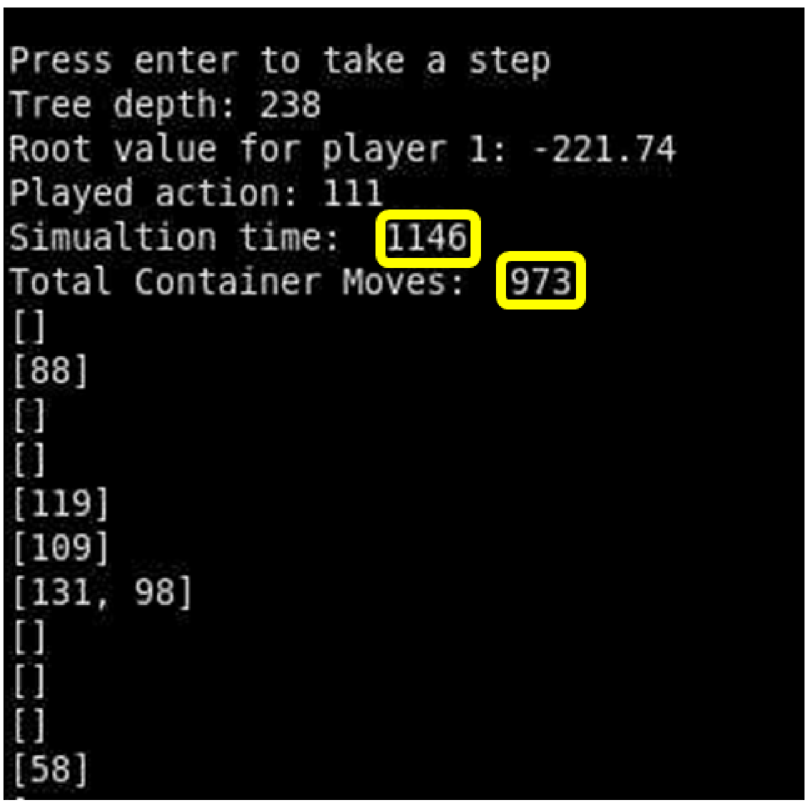

Figure 6-16. Demonstration of the algorithm choosing the “no move” action to reduce the number of touches per container

Another useful feature of the algorithm is the ability to show the expected value of a move. Consider another simulation using the same algorithm on a reduced size space. Figure 6-17 describes the change in root value from time step 74 to time step 75. At step 74, the algorithm calculates that the entire simulation has an expected reward of 172 moves – roughly 3.9 touches per container. At step 75, performance has improved – moving container 7 from stack two to stack seven reduced the number of expected moves by about six, reducing the expected total number of moves from 172 to 166, an improvement of about 0.14 touches per container. Note that this move created another empty stack – this highlights the value of an empty stack to the algorithm in quantitative terms. An individual may use this method to test new expert controls systems. This can also be leveraged to examine the impacts of disruptions and measure system resilience.

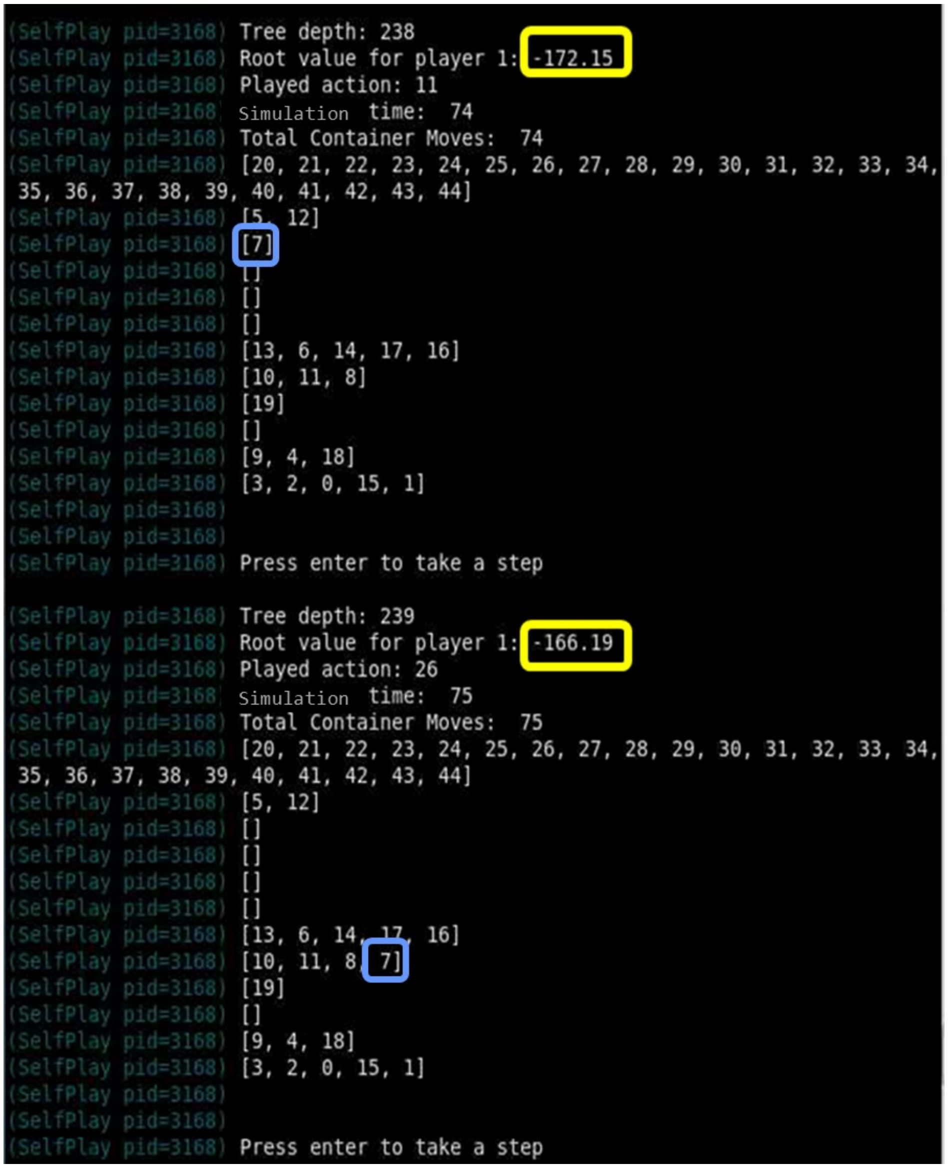

Figure 6-17. Demonstration of the algorithm providing the value of a move. In this case, the algorithm improved expected performance by six total moves.

6.7 MuZero and system resilience

In the previous section, MuZero was used to minimize the number of container touches and explore the impacts of changes to the container stacking block configuration. It was shown that a taller stacking block has similar performance to the baseline stack, while two additional stacks in the block improve performance by around 0.4 touches per container. This section observes how the algorithm responds to disruptive scenarios as outlined in Chapter 4.

Consider the following scenario – a hurricane off the coast of Virginia prevents vessels from reaching the port. Limited rains and flooding allow trucks to continue operations in the meantime. The result of this is many containers already in the container stacking block will leave the port before new arrivals. However, new arrivals will enter the system in a sharp wave. This sharp incoming wave is represented by a triangle distribution, which creates a period of no arrivals followed by a period of dense and frequent arrivals. Table 6-5 describes the performance of the non-disrupted scenario, the disruption scenario of the base model, and the performance of a specialized model trained on the disruptive scenario specifically. This will demonstrate the flexibility of the MuZero algorithm as a solution for improving system resilience to disruption.

The disrupted scenarios have two distinct phases – before the disruption begins (in this example around time step 100), and after. Prior to disruption, the container stack begins around 50% utilization, and this decreases until step 100 as containers exit the block. Lower utilization correlates with fewer touches per container, so prior to the arrival of containers, performance is better. When the disruption begins utilization can be raised to 90% and above, vastly increasing the number of touches required to handle containers.

Table 6-5. Sample of results comparing baseline model performance, performance under disruption, and performance using a model trained on disruption

Model	Arrivals	Utilization	Touches/container
Baseline	150 [U(0,1000)]	~70-80%	~4.6 – 5.6
Baseline (disrupted)	150 [Triangular (100,600,1200)]	~30-40% (pre-disruption) ~90% (disruption)	~3.5 – 4.0 (pre-disruption) ~10.2 (disruption)
Trained on disrupted	150 [Triangular (100,600,1200)]	~30-40% (pre-disruption) ~90% (disruption)	~4.5 – 5.5 (pre-disruption) ~10 (disruption)

Ultimately, the baseline algorithm has similar performance to the specialized model. The baseline model outperforms the specialized model in the non-disrupted portion of the simulation by about 1.0 touches per container, while the specialized model has better performance during disruption by roughly 0.2 touches per container. Ultimately, the two models had similar overall performance, with the base model around 3.5 – 4.0 touches per container in the non-disruption period and 10.2 touches per container in the disrupted period. The specialized model makes around 4.5 – 5.5 touches per container in the non-disrupted period and 10.0 in the disrupted period. It is notable that the model trained on the disrupted scenario has worse performance in the pre-disruption period. By observation, the model appears to make significantly more "wasted" moves, such as moving a container to a new stack in one time step before immediately moving it back in the next time step. This may reflect the increased uncertainty of the disrupted arrival distribution compared to the uniform baseline arrival distribution.

Similar heuristics to the baseline model are applied across both phases of the simulation, though it should be noted that the model trained on the disruptive scenario tends to rapidly empty stacks early in the simulation. Further, the model trained on the disrupted scenario goes to great lengths to avoid mixing stacks of incoming and pre-populated containers. This highlights the ability of the algorithm to react to uncertain future conditions – as noted above, the behavior of the two models is different in the non-disrupted periods. It should be noted that using both the baseline and specialized models, the container stack semi-regularly meets the early termination state (that is, there are no legal moves remaining). While the Port of Virginia has the flexibility to send containers to another stacking block or otherwise store surplus containers, the simulation does not have this capability.

The results of this simulation and the MuZero model are resilience techniques that form the new requirements for the system of interest for the CPSRM. These are the results of stages three, four, and five of the CPSRM. Updating the parameters of the algorithm to understand performance is a key part of the decision process between T2 and red team. The results gathered here are collected and sent to the blue team for stage six of the CPSRM.

Chapter 7: Mathematical framework for analysis of the disruption of system orders

7.1 Overview

This chapter describes the mathematical framework for disruption analysis and shows how it is integrated into the CPSRM. Section 7.2 describes the relevant background needed to understand the system using the mathematical framework. Section 7.3 describes the methodology for executing the mathematical framework. Section 7.4 describes three cases using the mathematical framework. Section 7.4.1 describes the first case analyzing a set of potential projects. Section 7.4.2 describes the second case analyzing the largest employers of the region. Section 7.4.3 describes the third case analyzing logistics assets. Section 7.4.4 presents a sensitivity analysis of the mathematical framework. Section 7.5 presents conclusions from the mathematical framework and concludes the CPSRM process.

This chapter represents stage six of the CPSRM. In stage *S6 – Accept System Description*, the blue team determines which resilience capabilities to incorporate based on the analysis by the T2 and red team. To accomplish this, the blue team utilizes a scenario-based mathematical framework to examine organizational priorities and assess how these priorities change due to disruption. Ultimately, the blue team finalizes the systems design, including the new and updated resilience techniques. This framework utilizes and extends a model applied to a sociotechnical system of systems (Loose *et al.* 2022(c), Loose *et al.* 2022(a)).

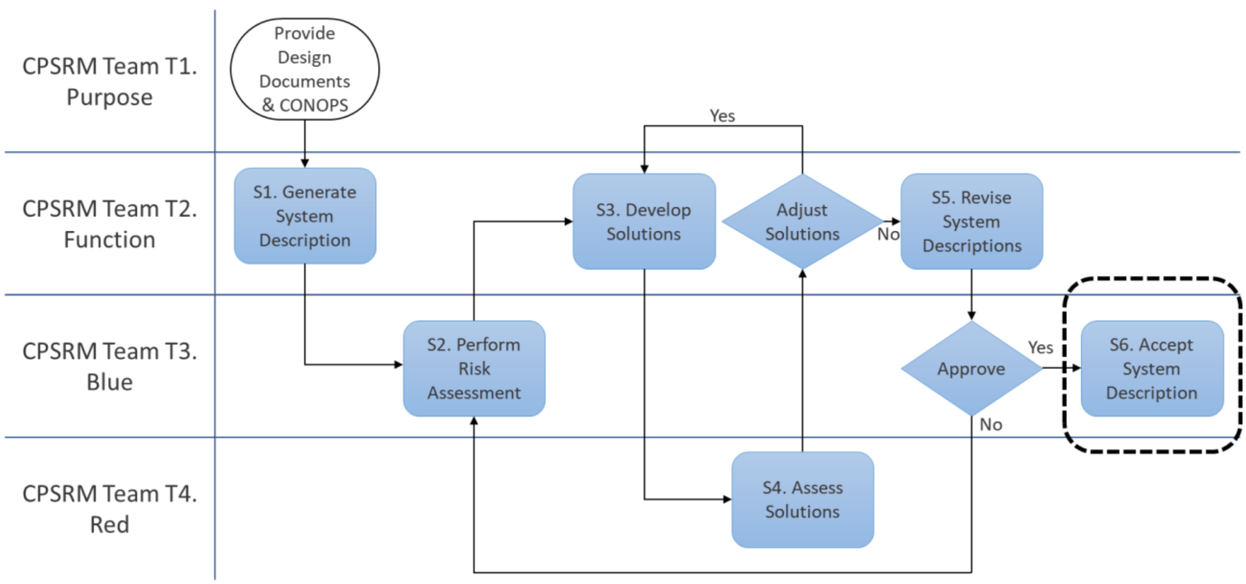

Figure 7-1. Relationship between CPSRM stages and Chapter 7 – Mathematical framework for analysis of the disruption of system orders. Chapter 6 describes stage six of the CPSRM, in which the blue team analyses the priority orders of the resilience capabilities outlined in previous steps. The team utilizes a scenario-based mathematical framework to collect stakeholder perspectives and priorities and provide insight into how orders change due to disruptive scenarios.

The system the blue team analyzes here is a large industrial region in Southeast Virginia, within the 500-mile radius of influence of the Port of Virginia. Stakeholders within the Port of

Virginia were consulted as experts of the region and were asked to assist in the development of priorities and in the analysis of results (Lambert *et al*. 2022(c), Lambert, Loose *et al.* 2022).

7.2 Background of the industrial region and relevance to the Port of Virginia

The decision maker for this analysis is the Crater Planning District Commission (CPDC), a regional planning agency that coordinates and provides policy recommendations for 11 member jurisdictions (VCC 2021). These includes seven counties (Charles City, Chesterfield, Dinwiddie, Greensville, Prince George, Surry, Sussex) and four cities (Colonial Heights, Emporia, Hopewell, Petersburg). The region services two large cities – Richmond to the north, and Newport News to the East. The region has a population of roughly 180,000 across 2,500 square miles of territory within the CPDC jurisdiction (Crater PDC 2021). The CPDC goals and objectives largely align with that of the Port of Virginia, which is headquartered East of the region in the Hampton Roads region. The Port of Virginia has several assets in or adjacent to the CPDC region, including the Richmond Marine terminal. Given this, the analysis of the CPDC will serve as a demonstration case of the CPSRM.

The CPDC is empowered by the state government, and "emphasizes transportation, economic and small business development, the environment, and serves as a convener for major military-related discussion among the region's communities" (CPDC 2021). The missions of the CPDC are:

1. Identifying interjurisdictional issues and opportunities, and establishing plans and policies to address these issues
2. Identifying mechanisms for local governments, the private sector, and non-profits to implement plans and policies

3. Promoting cooperation among state and local jurisdictions
4. Providing technical assistance and information services to member jurisdictions

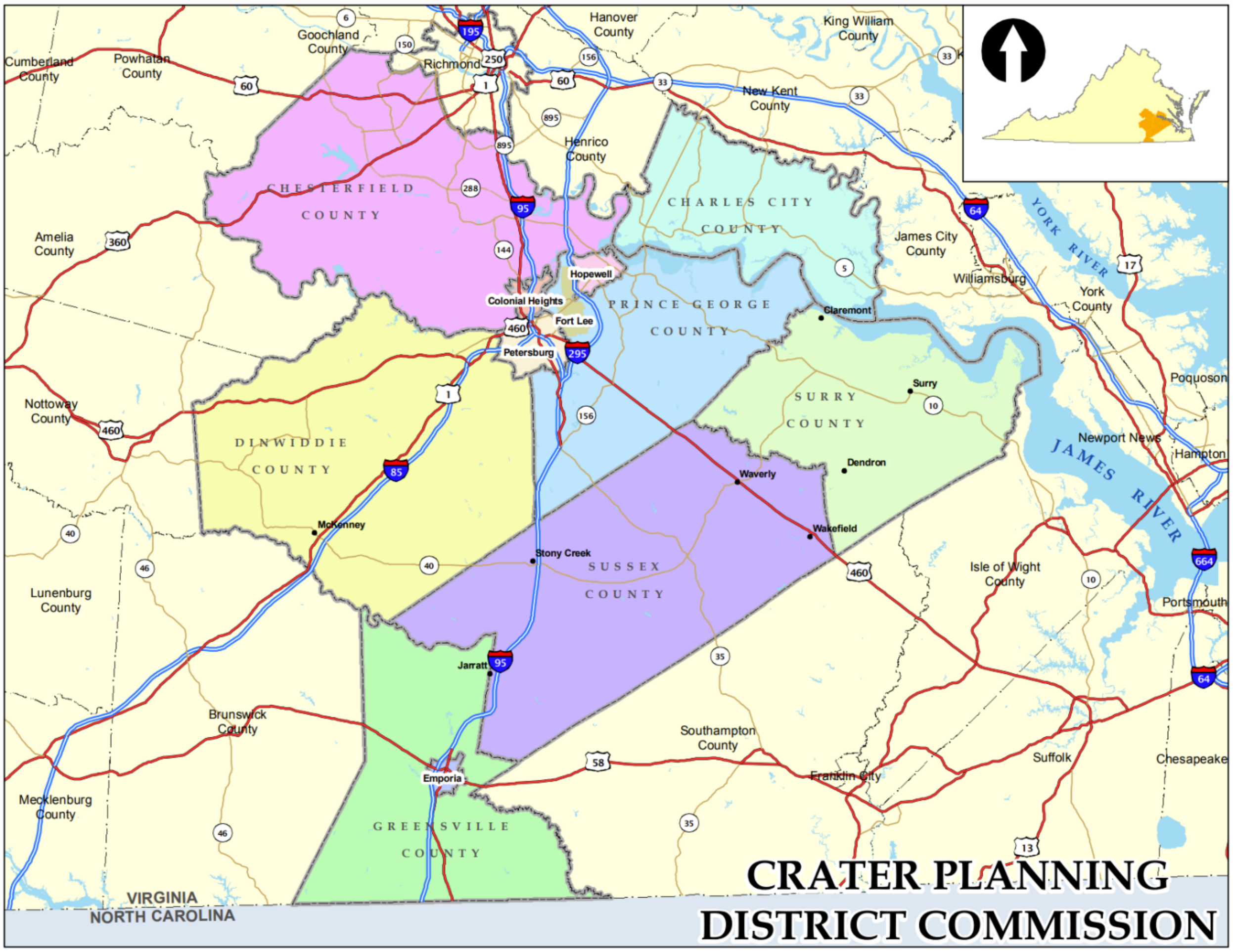

Figure 7-2. A map of the 11 member communities of the CPDC. The CPDC makes recommendations to improve regional government cooperation and planning to improve economic outcomes, safety, and sustainability (CPDC 2021).

The CPDC region hosts several facilities that are critical to the Port of Virginia, the state, and the nation as a whole. Interstates 95 and 85 pass through the region. Interstate 295 and Route 288 link directly to I-64, a critical roadway for the logistics systems of the state. With respect to the Port of Virginia, Routes 10, 58, and 460 all pass through the Crater region and connect the port to the district. The James River borders the CPDC region and links the Norfolk terminals of the port to inland terminals. Norfolk Southern and CSX railroads operate throughout the district, merging in the city of Petersburg and extending east to the Port. There are two major military installations within the CPDC as well – Fort Lee is contained entirely within the district, and is responsible for a large proportion of the economic activity within the region. About 10% of the economic activity in the Crater region is generated by Fort Lee (CPDC 2021). Fort Lee hosts "the U.S. Army Combined Arms Support Command/Sustainment Center of Excellence, the U.S. Army Quartermaster school, the U.S. Army Ordnance School, the U.S. Army Transpiration School, the Army Logistics University, Defense Contract Management Agency, and the U.S. Defense Commissary Agency" (U.S. Army 2021). These agencies and divisions are the heart of logistics training and operations for the U.S. Army, and rely on the supply chain infrastructure of the region. Fort Pickett, a National Guard installation, is partially contained within the region and serves as another major driver of economic activity in the region.

The public sector is the largest employer in the CPDC region, with over 27% of employed citizens working for a government agency. This includes the Department of Defense, state and federal agencies, local governments, state correctional facilities, and schools. 13.3% of working population is employed in healthcare including hospitals, clinics, and other medical facilities. Retail is the third largest employment sector with 13.2% of workers employed by organizations such Wal-Mart, Amazon, or Food Lion. The next largest sector is manufacturing, which accounts for 10.3% of employment in the region (USBLS 2021). The COVID-19 pandemic led to widespread job loss and a reduction in employment – at its peak in April 2020, unemployment reached 11.1%, down from the 2019 average of roughly 3%. This change represented 21,175 jobs lost (VEC 2021, Chmura 2021). This disruption, in conjunction with other disruptors such as unexpected budget changes and larger technological and demographic trends spurred the need for analysis of regional priorities with the assistance of the Port of Virginia as a key stakeholder.

It is critical for stakeholders in complex sociotechnical systems of systems, such as the CPDC and Port of Virginia, to understand the risks to their systems. Risk and uncertainty can be defined in several ways. Some organizations consider risk to be the loss of capabilities due to disruption across two metrics: the probability of disruption and the consequences of disruption (Conrow 2007). Other organizations consider a third metric – vulnerability, or the conditional probability that a system is damaged given a disruption has occurred. Still other definitions exist – the International Organization for Standardization considers risk to be "the effect of uncertainty on objectives" (ISO3100 2018). This dissertation, as well as the demonstration in Chapter 7, uses a similar perspective, extending the definition to be "the measurement of the influence of scenarios on priority orders" (Loose *et al.* 2023(a), 2022(c), 2021, Hassler 2020).

The mathematical framework presented in this chapter is used to understand risk to systems given the definition above. It is difficult to prioritize sets of projects, technologies, assets, and policies under standard conditions, especially in large and distributed systems such as the CPDC region. Disruptive scenarios such as the pandemic and natural disasters are a further compounding factor, making it more difficult to prioritize initiatives. The framework presented here enables analysts to organize their priorities such that they align with organizational goals and provide insight into how scenarios disrupt these priorities. This dissertation extends the framework by both defining how scenarios disrupt priorities and quantifying how different stakeholder perspectives influence priorities. This is accomplished by creating various stakeholder "profiles", which represent analysts with varying degrees of tolerance for risk. The framework calculates how the ranking of policies, projects, and assets change due to disruption. This results in two major outputs

– an examination of how the ranks of individual priorities rise and fall due to disruption, and an assessment of which scenarios are most disruptive to priority orders. This information is then be used by analysts to determine how to continue developing the system given uncertainty in future conditions.

7.3 Methodology of the mathematical framework

The mathematical framework is used to assist analysts with understanding what potential future conditions are most disruptive to system priorities and how priorities change due to these disruptions. To utilize the framework, analysts – in this case the blue team in the CPSRM framework – execute several steps to provide inputs, calculations, and generate outputs. Figure 7-3 provides a graphical overview of the framework. Table 7-1 provides an overview of data used for the mathematical framework.

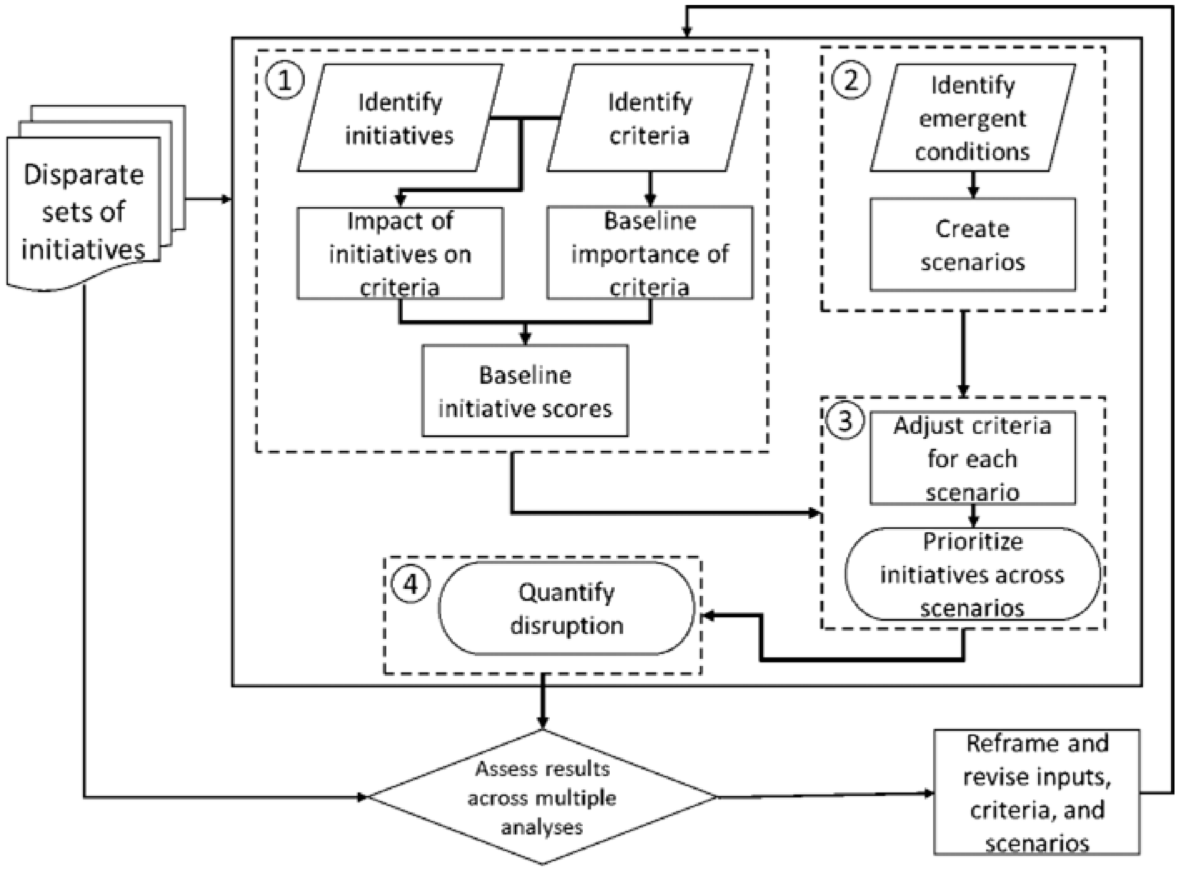

Figure 7-3. The steps of the mathematical framework. The framework is one of the tools used in the CPSRM, assisting the blue team with selecting resilience measures based on how priorities change under disruption. The framework can also be utilized by T2 to identify the most disruptive scenarios. Adapted from (Loose *et al.* 2022(c), 2023(a)).

Table 7-1. Data for framework. Descriptions and notations of the data used in the framework (adapted from Loose *et al.* 2023(a)).

Data	**Notation**	**Description**
Success Criteria	$C = \{c_1, ..., c_m\}; 1 \leq j \leq m$	The set of success criteria
Initiatives	$X = \{x_1, ..., x_n\}; 1 \leq i \leq n$	The set of initiatives
Emergent Conditions	$E = \{e_1, ..., e_q\}; 1 \leq l \leq q$	The set of emergent conditions
Scenarios	$S = \{s_0, ..., s_p\}; 0 \leq k \leq p$	The set of disruptive scenarios, where s_0 is the baseline scenario
Weight Matrix	$W_{p \times m}$	The set of importance scores for each success criterion c_j across scenarios s_k
Importance Matrix	$V_{p \times n}$	The set of value functions for each initiative x_i with each criterion c_j
Rank Matrix	$R_{p \times n}$	The set of ordered priorities for each initiative x_i across each scenario s_k
Disruptiveness Vector	$D = \{d_0, ..., d_p\}; 0 \leq k \leq p$	The set of disruptiveness scores across each scenario s_k

In the first step the blue team develops success criteria and sets of initiatives for analysis. Also called figures of merit, success criteria are used to identify the metrics, values, and missions of the organization, in this case the CPDC. Table 7-2 shows the success criteria used in this analysis. The success criteria $C = \{c_1, ..., c_m\}$ are derived from several sources, including third-party program analysis, literature reviews, established standards, internal expertise, and other sources.

Table 7-2. Success Criteria. Set of success criteria used across all demonstrations of the mathematical framework (adapted from Loose *et al.* 2023(a))

Index	Criterion	Description
c_1	Quality of Life	Ability of citizens to comfortably participate in life events
c_2	Innovation	Development of new ideas, products, or policies
c_3	Economic Development	A measure of regional prosperity such as regional GNP
c_4	Economic Resilience	The ability to withstand disruptive scenarios
c_5	Carbon Footprint	Total greenhouse gas emissions due to activities
c_6	Sustainability	Ability to maintain ecological balance across operations
c_7	Safety	Protection from or reduction in risk of injury or other harm

Success criteria are used to rank and order the sets of initiatives. The following provides greater detail regarding the success criteria used throughout this analysis. While multiple sets of initiatives are used in this chapter, the performance criteria used by the blue team remains constant. Success criteria are defined during stage 1 of the CPSRM, in which T2 creates a system description based on information provided in design documents and through system analysis. c_1 – *Quality of life* is a measure of the degree to which an initiative impacts the standard of living of the citizens. Changes to c_1 – *quality of life* include increased wages, the creation of new jobs, introducing new technologies to the population, providing critical or luxury services, or the incorporation of new amenities. c_2 – *Innovation* is the degree to which an initiative generates new ideas, technologies, methods, or products that are beneficial to the system or can be used to enhance other criteria. c_3 – *Economic development* is the ability of an initiative to improve economic metrics such as job growth, employment rates, infrastructure investment, or the development of partnerships which attract new organizations, employers, or investors to the region. c_4 – *Economic resilience* refers to the degree to which an initiative protects the region from fluctuations in broader economic conditions. c_5 – *Carbon footprint* is a measure of the change in carbon emissions of the region due to the implementation of an initiative. c_6 – *Sustainability* is the ability of an initiative to address current conditions and requirements while reducing the negative impacts to future generations, especially regarding the environment, climate change, and equity. c_7 – *Safety* is the degree that an initiative improves or maintains the safety of the population, primarily addressing physical harm or trauma.

The blue team determines the importance of each criterion under normal conditions (the *baseline*), rated *high, medium*, or *low* importance, with a weight associated with each. Later sections of this chapter will discuss the influence of these weights on results, but the initial set of weights is four for *high,* two for *medium* and one for *low*. The framework utilizes discrete weights rather than continuous as it has been shown that ordinal ranking is faster and more efficient than cardinal methods, while maintaining a similar level of accuracy (Ali and Ronaldson 2012). The weights are stored in the weight matrix *W*.

Table 7-3. Success Criteria Weights. The set of weights used in the initial assessments

Criterion Weight	Numerical Weight
High	4
Medium	2
Low	1

Table 7-4. Criterion ratings for CPDC analysis. The rankings *high, medium,* and *low* used throughout the case studies. Note that the analysts in the CPDC did not rank any criteria as *low.*

the criterion c.xx has	-	relevance among the other criteria
c.01 - Quality of Life has	high	relevance
c.02 - Innovation has	medium	relevance
c.03 - Economic Development has	high	relevance
c.04 - Economic Resilience has	high	relevance
c.05 - Carbon Footprint has	high	relevance
c.06- Sustainability has	medium	relevance
c.07 - Safety has	medium	relevance

Initiatives are technologies, policies, projects, assets, and other system components or investments that are implemented to impact the success criteria. The set $X = \{x_1, ..., x_n\}$ of initiatives are derived from stakeholder input, system experts, and the revised system description from T2 and red team from stage five of the CPSRM. The new resilience techniques and technologies developed in steps one through five are inputs to the mathematical framework in step six. Analysts on the blue team are asked to respond to the statement "initiative x_i is relevant to the criterion c_j", responding as *strongly agree, agree, somewhat agree*, or *neutral*, corresponding with initial weights of 1, 2/3, 1/3, and 0 respectively. Within the framework itself, these values are represented by a filled circle (●), a half-filled circle (◐), an empty circle (○), and a hyphen (▭). Table 7-5 describes this relationship. This process is called the criteria-initiative (C-I) assessment. As an example, consider the success criterion *quality of life*, and an initiative such as *waste management*. As *waste management* has high relevance to *quality of life*, the blue team would choose *strongly agree*.

Table 7-5. Criteria-Initiative Weights. The values and representation symbols for the baseline analysis of criteria and initiative

Representation	Weight	Criteria-Initiative Assessment
●	1	strongly agree
◐	0.667	agree
○	0.333	somewhat agree
▬	0	neutral

In step two of the framework, the blue team inputs emergent conditions and generates scenarios. These are taken from step two of the CPSRM, in which the blue team performs a risk assessment to identify sources of risk to the system such as natural and human-caused hazards. Emergent conditions are future events, trends, or other uncertain factors that impact the effectiveness of initiatives with respect to their impact on success criteria. The set $E = \{e_1, ..., e_i\}$ of emergent conditions is derived from literature, analysis of the system, stakeholder experience, and forecasts of future conditions. Emergent conditions are combined to form scenarios. That is, scenarios consist of one or more emergent conditions. The set $S = \{s_0, ..., s_j\}$ of scenarios is a series of potential events that may disrupt priority orders. The scenario s_0 is the *baseline* scenario in which no emergent conditions arise – also called the *status quo* or as-planned circumstances. It should be noted that scenarios are not predictions of future conditions and carry no notion of probability of occurrence. Scenarios are projections used to explore the effects of the potential future states. Further, emergent conditions and scenarios do not attempt to enumerate all possible future states or disruptions, but focus on the concerns of the system owners and analysis such as the CPDC, the Port of Virginia, or the Virginia Department of Emergency Management.

The influence of scenarios to priorities is seen through how scenarios influence the relative importance of success criteria. That is, the priorities of system owners change when the system is exposed to disruption. In step three of the framework, the blue team adjusts the importance of the criteria across each scenario. An analyst is asked to determine if "the relative importance of criterion c_j" *increases, increases somewhat, does not change, decreases somewhat,* or *decreases* "for scenario s_k when compared to the baseline s_0". Each answer has a corresponding weight, in this case 8, 6, 1, 1/6, and 1/8 respectively. The responses are stored in the matrix W across each of the k scenarios.

These weights can be changed in subsequent analyses based on user preferences. This assessment is based on expertise, institutional knowledge, and through iteration with other experts. Consider the example above – the *waste management* initiative and *quality of life* criterion. Under a *natural disaster* scenario, the relative importance of the *quality of life* criterion rises. As the *waste management* initiative had a high impact on *quality of life,* it may rise in priority as the weightings of criteria change. This process is explored in more detail across the three demonstration cases, and the impacts of weights are explored in the sensitivity analysis.

Table 7-6. Weights for analyst responses to Criteria-Scenario analysis

Criteria - Scenario Importance Change	Weight
Increases	8
Increases Somewhat	6
-	1
Decreases Somewhat	0.1667
Decreases	0.125

This information is used to give each initiative a score across each scenario. Equation 7.1 describes the linear additive value function used to generate these scores.

$$V_k(x_i) = \sum_{j=1}^{m} w_{jk} v_j(x_i) \tag{7.1}$$

$v_j(x)$ is the partial value function for initiative x_i for criterion c_j as defined in the criteria-initiative assessment. V is a matrix that contains the relative importance scores for each initiative across each scenario. Given these scores, the initiatives are ordered for each scenario including the baseline s_0 and added to the matrix R, the rank matrix. In R, r_{ik} represents the priority of an initiative x_i for scenario s_k. Each initiative score is ranked for each scenario, providing a ranking from 1 to n. Equation 7.2 describes this process, where $\succ$ indicates that an initiative has a higher ordinal ranking. That is, if the score of an initiative x_i is higher than an initiative x_a, than it has a higher ordinal ranking – for example, they may be ranked 1 and 2 respectively.

$$IF\ V(x_i)_k > V(x_a)_k\ THEN\ x_i \succ x_a \tag{7.2}$$

In step four of the framework, the blue team develops disruptiveness scores for each scenario. Disruptiveness is a measure of the degree to which priority orders change under a given scenario. The sum of squares differences between an initiative in the baseline and disrupted scenario is sued to find the disruptiveness of each scenario. Equation 7.3 describes the disruptiveness measure D_k.

$$D_k = \frac{\sum_{i=1}^{n}(r_{i0} - r_{ik})^2}{\sum_k D_k} \tag{7.3}$$

The disruptiveness D_k of scenario s_k is equal to the sum of squares difference in priority for each initiative when compared to the baseline scenario. These scores are normalized so they can be compared to one another easily.

7.4 Demonstrations of the mathematical framework

The inputs to the mathematical framework are derived from the previous five steps of the CPSRM. Initiatives are taken from the system design documents and stakeholders, emergent conditions from the hazard analysis. This section outlines three sets of analyses created for the CPDC, with inputs from the Port of Virginia and federal and state governments – case one is an assessment of projects that the CPDC controls; case two is a list of the largest employers of the region, which the CPDC coordinates with to improve economic outcomes; case three is an analysis of the major logistics assets of the region, including roads, railroads, warehouses, utility services, and other industrial assets. Analysis is separated for two reasons – first, it is difficult to compare the importance of disparate assets (such as comparing an existing roadway to the importance of a new business), and second breaking up the analysis enables the blue team to isolate how different disruptive scenarios impact different segments of the system in isolation. Determining which initiatives are susceptible to disruption and which scenarios are the most disruptive will help the CPDC and the port to understand their system, understanding how the system may react to disruption in the future. These analyses represent the types of outputs systems analysts can expect from performing the CPSRM.

Each of the three analyses utilize the same sets of criteria, emergent conditions, and scenarios. Table 7-7 describes the set of emergent conditions used across each case. Figures 7-8 and 7-9 describe the formation of scenarios and an overview of the scenarios used in the analysis. Figure 7-10 outlines the criteria-scenario analysis, showing how criterion priorities change under each scenario.

Table 7-7. Emergent Conditions. Emergent conditions are potential future states that impact the priorities of the system. The set of emergent conditions here used across all three cases.

Index	Emergent Condition
e_1	Social Distancing Requirements
e_2	Fiscal Stress
e_3	Lack of Infrastructure
e_4	Lack of Well-Trained Workforce
e_5	Reduced Regional Cooperation
e_6	Poor Health Indicators
e_7	Funding Decrease
e_8	Flooding
e_9	Changes to Economic Systems
e_{10}	Category 5 Hurricane
e_{11}	Decreased Social Cohesion
e_{12}	Multiple Super Storms in Same Season
e_{13}	Increased Operational Costs
e_{14}	Increased Emphasis on Emissions
e_{15}	Decrease in Labor Productivity
e_{16}	Drought
e_{17}	Population Increase
e_{18}	Population Decrease
e_{19}	Social Unrest
e_{20}	Network Damage due to Cyber Attack
e_{21}	Ransomware Attack
e_{22}	Data Leak
e_{23}	Decreased Security of Vital Infrastructure
e_{24}	Destruction of Infrastructure
e_l	Others

Table 7-8. Scenario formation. Scenarios are made up of one or more emergent conditions. This figure shows which emergent conditions make up each of the scenarios in the CPDC jurisdiction.

Scenario	e.01 - Social Distancing Requirements	e.02 - Fiscal Stress	e.03 - Lack of Infrastructure	e.04 - Lack of Higher-Skilled, Well-Trained Workforce	e.05 - Reduced Regional Cooperation	e.06 - Poor Health Indicators	e.07 - Funding Decrease	e.08 - Flooding	e.09 - Changes to Economic Systems	e.10 - Category 5 Hurricane	e.11 - Decreased Social Cohesion	e.12 - Multiple Super Storms in Same Season	e.13 - Increased Operational Costs	e.14 - Increased Public Emphasis on Carbon Emissions	e.15 - Decrease in Total Labor Productivity	e.16 - Drought	e.17 - Population Increase	e.18 - Population Decrease	e.19 - Social Unrest	e.20 - Network Damage due to Cyber Attack	e.21 - Ransomware Attack	e.22 - Data Leak	e.23 - Decreased Security of Vital Infrastructure	e.24 - Destruction of Infrastructure
01 Funding Decrease		x			x		x				x						x	x					x	x
02 Natural Disaster		x			x			x		x	x	x	x	x	x				x					x
03 Pandemic	x	x		x	x	x					x		x					x	x					
04 Increased Environmental Regulation				x	x				x				x											
05 Climate Shift		x						x		x		x	x	x	x	x		x						x
06 Green Technology Movement			x	x										x										
07 Cyber Security Attack		x																		x	x	x		
08 Others																								

Table 7-9. Scenarios. The set of scenarios that disrupt priorities, formed from emergent conditions. These scenarios are used across each of the three cases.

Index	**Disruptive Scenario**
s_0	Baseline (No Disruption)
s_1	Funding Decrease
s_2	Natural Disaster
s_3	Pandemic
s_4	New Environmental Regulation
s_5	Climate Shift
s_6	Green Technology Movement
s_7	Cyber Security Attack
s_k	Others

Table 7-10. Relative importance score changes. When scenarios disrupt systems, the relative importance of success criteria may change based on which system components are impacted. The relative importance changes for each criterion and each scenario are outlined here.

p.01 Crater PDC	s.01 - Funding Decrease	s.02 - Natural Disaster	s.03 - Pandemic	s.04 - Increased Environmental Regulation	s.05 - Climate Shift	s.06 - Green Technology Movement	s.07 - Cyber Security Attack
c.01 - Quality of Life	-	-	*Increases*	-	-	-	-
c.02 - Innovation	*Increases*	-	-	*Increases Somewhat*	*Increases*	*Increases*	-
c.03 - Economic Development	*Decreases Somewhat*	*Decreases Somewhat*	*Decreases*	-	-	*Increases Somewhat*	-
c.04 - Economic Resilience	*Increases*	-	*Decreases Somewhat*	-	*Increases Somewhat*	-	*Increases Somewhat*
c.05 - Carbon Footprint	-	*Increases*	*Decreases*	-	*Increases Somewhat*	*Increases*	*Decreases*
c.06 - Affordability	*Increases*	*Increases Somewhat*	*Increases Somewhat*	-	*Decreases Somewhat*	*Decreases Somewhat*	-
c.07 - Sustainability	-	-	*Increases*	-	*Increases*	*Increases Somewhat*	-
c.08 - Feasibility	*Increases Somewhat*	*Increases*	-	-	*Decreases Somewhat*	-	*Decreases*
c.09 - Safety	-	-	*Increases*	-	*Increases*	*Increases Somewhat*	*Increases*

7.4.1 Case One: CPDC Projects

The first case utilizes a list of 21 projects that the CPDC controls, sourced from the CPDC economic development strategy (Crater PDC 2021). Many of these projects directly influence major infrastructure such as improvements to highways, and are of immediate interest to the Port of Virginia and its 500-mile region of influence. The projects receive funding from state, federal, and private entities. Table 7-11 describes the set of initiatives. Table 7-12 gives an overview of the criteria-initiative assessment completed for case one.

Table 7-11. The set of initiatives used in case one: CPDC projects. Derived from the Crater Economic Development Strategy (Crater PDC 2021)

Index	Initiative
x_1	MAMaC 1,600 Acre Mega Site
x_2	Sussex County Route 626 1,500 Acre Mega Site
x_3	Global Logistics Park
x_4	I-95 / I-85 Interchange Improvements
x_5	Appomattox River Dredging Project
x_6	Redevelopment of Exit 52 on I-95
x_7	Crater Small Business Development Center
x_8	Crater Procurement Technical Assistance Center
x_9	Improvements to Existing U.S. Route 460
x_{10}	New Raw Water Intake and Waterline
x_{11}	Grey's Creek Marina Project
x_{12}	Route 602 (Cabin Point Road) Industrial Park
x_{13}	Water Line Extension - Mega Site
x_{14}	Dendron Area Water System Replacement
x_{15}	Route 36 Regional Corridor Revitalization Project
x_{16}	Halifax Industrial Park Site Improvements
x_{17}	Business Incubator
x_{18}	Claremont Water System Improvements
x_{19}	Stony Creek Wastewater Treatment Plant Upgrade
x_{20}	Tri-Cities Area Business Incubator
x_{21}	New Industrial Property - 38 Acres
x_i	Others

Table 7-12. Case one criteria-initiative assessment. The unique criteria-initiative assessment for case one: CPDC projects. Ratings are represented by a filled circle (●, strongly agree), a half-filled circle (◐, agree), an empty circle (○, somewhat agree), and a hyphen (━, neutral)

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
x.01 - MAMaC 1,600 Acre Mega Site	●	◐	●	●	●	━	━
x.02 - Sussex County Route 626 1,500 Acre Mega Site	●	━	●	●	━	━	━
x.03 - Global Logistics Park	○	●	●	◐	━	◐	●
x.04 - I-95 / I-85 Interchange Improvements	●	━	◐	━	◐	○	◐
x.05 - Appomattox River Dredging Project	◐	○	━	○	●	●	○
x.06 - Redevelopment of Exit 52 on I-95 (City of Petersburg Gateway)	●	━	◐	◐	━	○	○
x.07 - Crater Small Business Development Center	●	◐	●	●	━	○	◐
x.08 - Crater Procurement Technical Assistance Center	○	◐	●	◐	○	◐	○
x.09 - Improvements to Existing U.S. Route 460	●	━	◐	○	○	○	●
x.10 - Raw water intake and waterline to Roxbury Area of Charles City County	◐	◐	━	━	○	●	○
x.11 - Grey's Creek Marina Project, public access to James River	○	◐	●	●	○	○	◐
x.12 - Route 602 (Cabin Point Road) Industrial Park - 134 Acres	○	◐	●	●	━	━	○
x.13 - Water Line Extension - Mega Site	◐	○	○	○	●	◐	━
x.14 - Dendron Area Water System Replacement	◐	●	○	◐	●	●	◐
x.15 - Route 36 Regional Corridor Revitalization Project	●	◐	●	●	━	◐	○
x.16 - Halifax Industrial Park Site Improvements (one site)	○	●	◐	◐	○	○	◐
x.17 - Business Incubator Collocated w/ Southside Virginia Educational Center	●	●	●	●	○	○	◐
x.18 - Claremont Water System Improvements	◐	○	○	○	●	○	◐
x.19 - Stony Creek Wastewater Treatment Plant Upgrade	●	○	○	◐	●	●	○
x.20 - Tri-Cities Area Business Incubator	●	○	○	◐	●	◐	◐
x.21 - New Industrial Property - 38 Acres	○	●	●	◐	○	○	◐

Given these inputs, the framework generates two primary artifacts – an analysis of the initiatives, describing how they change in priority under disruption, and an analysis of the disruptiveness of each scenario. Figure 7-4 describes the first artifact, the analysis of the priority of initiatives. Table 7-13 shows the underlying data used to create the first artifact.

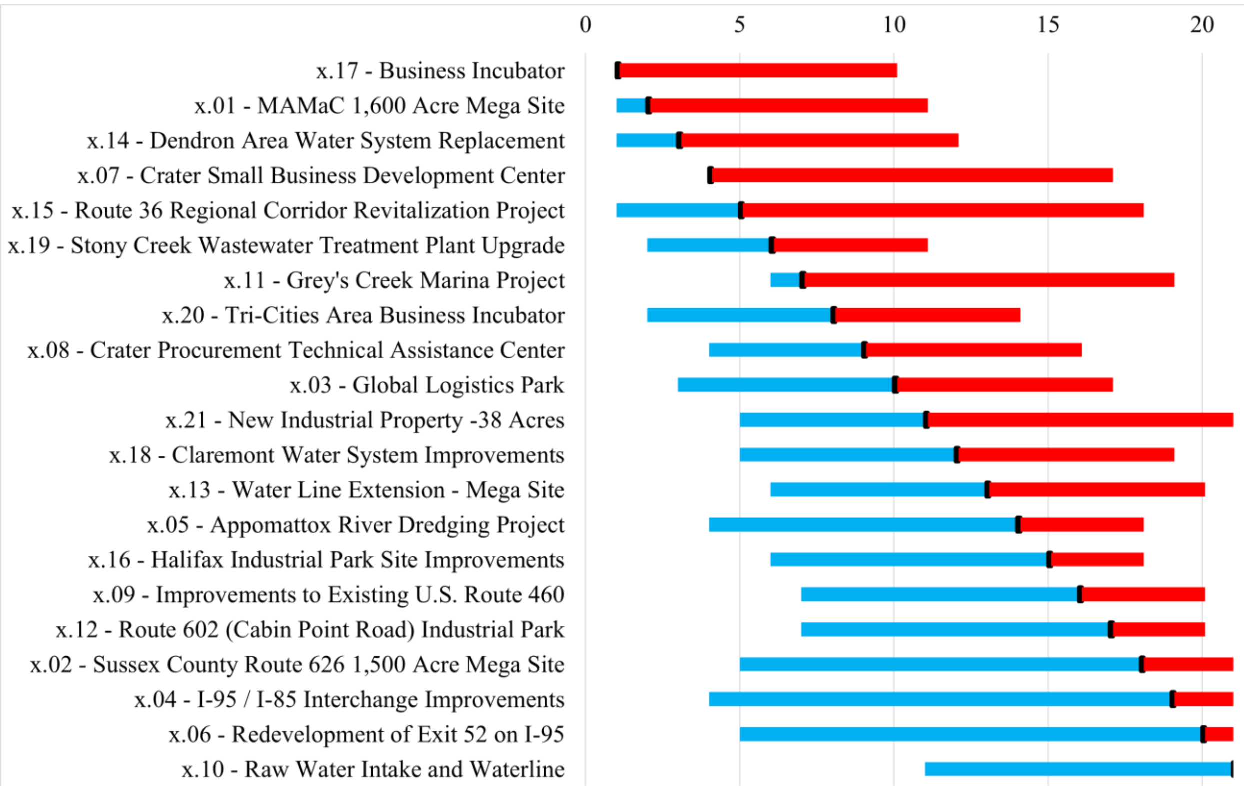

Figure 7-4. Results of mathematical framework for case one. The results of the framework are presented here, utilizing 21 projects outlined by the CPDC. The mark between the blue and red lines represents the priority of an initiative in the baseline scenario. The red bar shows how the priority may drop in priority due to disruption. The blue bar shows how an initiative rises in priority due to disruption.

Table 7-13 Initiative ranking chart. This table describes the ranking of each initiative under each scenario for the CPDC Projects analysis, the rank matrix $R_{p \times n}$. The green filled cells indicate a higher ranking. The red and orange filled cells indicate a lower ranking. This information is used to create the first artifact of the mathematical framework.

	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
x.01 - MAMaC 1,600 Acre Mega Site	2	11	7	10	3	1	1	3
x.02 - Sussex County Route 626 1,500 Acre Mega Site	18	15	21	14	20	17	19	5
x.03 - Global Logistics Park	10	3	15	17	4	16	12	8
x.04 - I-95 / I-85 Interchange Improvements	19	21	8	4	19	18	13	20
x.05 - Appomattox River Dredging Project	14	14	4	8	15	8	14	18
x.06 - Redevelopment of Exit 52 on I-95 (City of Petersburg Gateway)	20	17	19	5	21	19	20	12
x.07 - Crater Small Business Development Center	4	5	17	6	7	10	15	4
x.08 - Crater Procurement Technical Assistance Center	9	12	16	16	10	9	4	9
x.09 - Improvements to Existing U.S. Route 460	16	19	9	7	18	20	18	16
x.10 - Raw water intake and waterline to Roxbury Area of Charles City County	21	18	13	11	17	21	21	21
x.11 - Grey's Creek Marina Project, public access to James River	7	6	11	19	9	6	6	6
x.12 - Route 602 (Cabin Point Road) Industrial Park - 134 Acres	17	13	20	20	13	14	17	7
x.13 - Water Line Extension - Mega Site	13	20	6	13	16	12	7	17
x.14 - Dendron Area Water System Replacement	3	1	1	12	2	3	3	10
x.15 - Route 36 Regional Corridor Revitalization Project	5	3	18	1	7	10	16	2
x.16 - Halifax Industrial Park Site Improvements (one site)	15	8	12	18	6	7	11	15
x.17 - Business Incubator Collocated w/ Southside Virginia Educational Center	1	2	10	3	1	2	2	1
x.18 - Claremont Water System Improvements	12	16	5	15	14	15	10	19
x.19 - Stony Creek Wastewater Treatment Plant Upgrade	6	7	3	2	11	5	9	11
x.20 - Tri-Cities Area Business Incubator	8	10	2	9	11	4	8	14
x.21 - New Industrial Property - 38 Acres	11	9	14	21	5	13	5	13

Consider the results in Figure 7-4. If the baseline ranking of a scenario is centered on a wide bar, the initiative is sensitive to disruptions and does not rank consistently under disruptive scenarios. If the baseline falls on the left side of the bar with a long red segment, the initiative falls in priority due to one or more disruptions. If the initiative falls on the right side with a large blue segment, the opposite is true and the initiative is likely to rise in importance due to disruptions. Decision makers on the blue team can use this information when determining which initiatives to select for the final system description, altering their choices based on their beliefs about potential future conditions.

It is useful to observe how the priorities change under various scenarios as an aspect of enterprise risk analysis. The highest rated initiative in the baseline scenario, *x.17 – business incubator in southside Virginia educational center*, may fall as low as tenth in the *natural disaster* scenario, but remains highly ranked across the other scenarios. This initiative is relatively resistant to disruption. Initiatives such as *x.07 – Crater small business development center* rank highly in the baseline but can fall sharply under multiple scenarios – in this case both the *natural disaster* and *green technology movement* scenarios.

Initiatives of interest to the port include *x.06 – redevelopment of exit 52 on I-95* and *x.04 – I-95 /I-85 interchange improvements*. These projects directly impact two of the major roadways that service the port. *x.06* has a baseline ranking on the right edge of the bar, ranking 20th in the baseline scenario. In the *pandemic* scenario, the importance of *x.06* rises to fifth. Throughout the COVID-19 pandemic, logistics systems slowed and stalled, preventing many goods from being distributed. Long delays even led to container ships anchoring away from port for days at a time. The importance of functional, efficient roadways rose for the CPDC under a *pandemic* scenario to ensure goods were able to be distributed. A similar effect is seen in *x.04*, ranked 19th in the baseline but rising to 4th in the *pandemic* scenario and 8th in the *natural disaster* scenario. The reasoning is similar – during disruptive scenarios the importance of road availability for the movement of goods is increased. This information is used by the blue team to inform their recommendations for implementation of resilience priorities.

Figure 7-5 describes the normalized disruptiveness scores for case one: CPDC projects. A high disruptiveness score indicates that a scenario has a high influence on the disruption of priorities. The two most disruptive scenarios are *s.02 – natural disaster* and *s.03 – pandemic*, with *pandemic* being slightly more disruptive. Events such as the COVID-19 pandemic tend to raise or depress the relative importance of scenarios. As an example, *x.06 – redevelopment of exit 52 on I-95* rises in priority under the *pandemic* scenario. Some initiatives fall, such as *x.21 - New Industrial Property - 38 Acres*, which falls from priority 11 to priority 21 under the *pandemic* scenario. The *pandemic* scenario may delay infrastructure improvement projects as it prevents people from gathering due to social distancing requirements or keeping workers at home due to illness. *Natural disaster* scenarios may prevent the movement of people or goods along infrastructure pathways such as highways, make roads and buildings unsafe, or force segments of the population to temporarily or permanently move away from the region. Projects such as *x.15 - Route 36 Regional Corridor Revitalization Project* are particularly affected by the *natural disaster* scenario, falling from 5th to 18th in priority rankings. This is an expected result as *pandemics* and *natural disasters* occur without warning and can cause widespread (rather than siloed) disruption, while the other scenarios such as *climate shift* and *green technology movement* have longer time horizons and may be predictable. The exception to this is *s.07 – cyber security attack*. An attack such as the ransomware attack on the Colonial Pipeline may disrupt multiple projects at once (Turton and Mehotra 2021). This is also an abrupt and unpredictable event. However, such an attack is most likely to be isolated to one or a few initiatives and would not have the widespread impact of a *natural disaster* or *pandemic* scenario. Further, the damage caused by a cyber-attack is unlikely to be permanent whereas a natural disaster can permanently alter landscapes, assets, and other structures beyond repair. Table 7-14 describes an overview of the results of the analysis for case one.

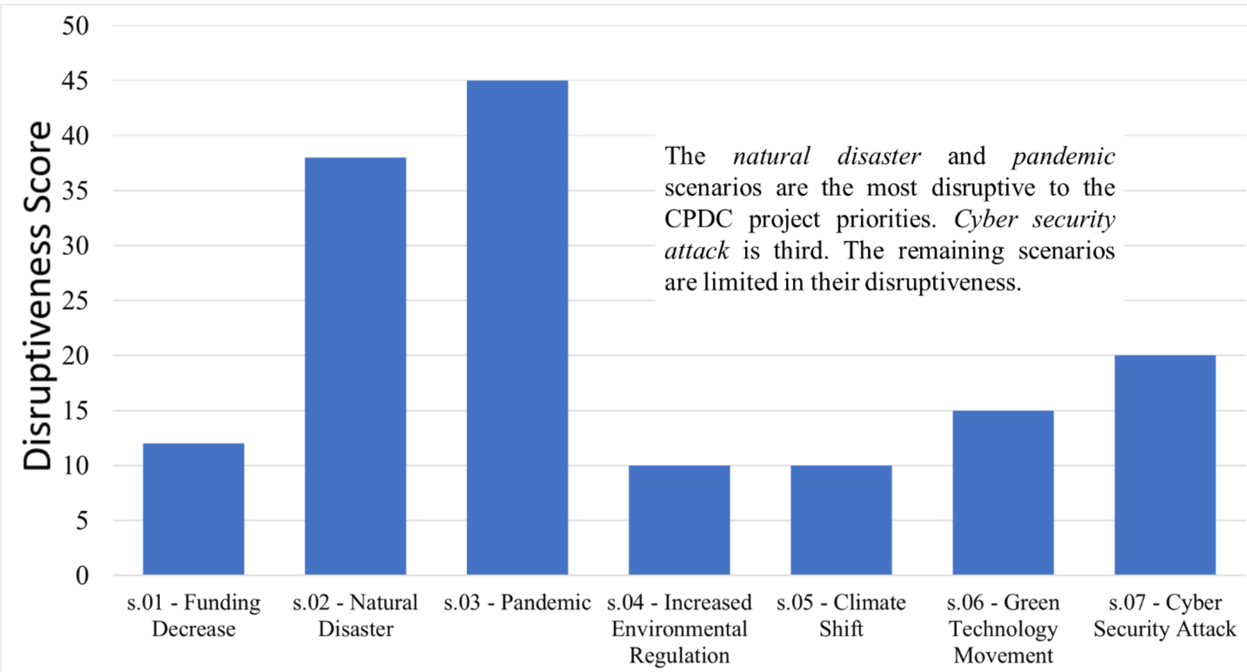

Figure 7-5. Disruptiveness scores of scenarios for CPDC projects. The figure presents the normalized disruptiveness scores across each scenario for the 21 CPDC projects. *Pandemics* and *natural disasters* are the most disruptive.

Table 7-14. Summary of results for case one: CPDC projects

Result - CPDC Projects	Description
Most resilient initiative	The initiative x_{17} - *Business Incubator* is one of the most resilient initiatives for the project analysis. It is ranked highly in the baseline and does not fall below rank ten.
Resilient initiatives	x_2, x_4, and x_6 are resilient initiatives. Though they rank low in the baseline, they have the potential to rise into the top five.
Most disruptive scenarios	s_2 - *Natural Disaster* and s_3 - *Pandemic* are the most disruptive scenarios, with s_3 - *Pandemic* slightly more disruptive. These are followed by s_7 - *Cyber Security Attack*.
Least disruptive scenarios	s_4 - *Increased Environmental Regulation* and s_5 - *Climate Shift* are tied as the least disruptive scenarios.

7.4.2 Case Two: CPDC Employers

Table 7-15 describes the 30 largest employers of the CPDC by total employees. The CPDC and Port of Virginia are interested in this set of initiatives as employers 1) provide jobs that improve economic outcomes for the regions and 2) deliver goods, services, and amenities to residents of the region. The CPDC benefits from understanding how various employers are impacted by disruption. The public sector is the largest employer of the region with employees in the US Department of Defense, schools, cities, counties, and other government organizations. Other large employers include healthcare organizations such as hospitals, clinics, and elder care facilities. Other major employers are private retailers and distribution organizations. Manufacturing employs many citizens of the region. Table 7-16 describes the criteria-initiative assessment completed for case two.

Table 7-15. List of the 30 largest employers of the CPDC region

Index	Initiative
x_1	US Department of Defense
x_2	Wal-Mart
x_3	County of Prince George
x_4	Dominion Energy
x_5	Central State Hospital
x_6	Boar's Head Provisions Co.
x_7	Integrity Staffing Solutions
x_8	Greensville Correctional Center
x_9	AdvanSix, Inc.
x_{10}	City of Petersburg
x_{11}	City of Petersburg School Board
x_{12}	Hopewell City School Board
x_{13}	Dinwiddie County School Board
x_{14}	HCA Virginia Health System
x_{15}	Food Lion
x_{16}	Good Neighbor Holdings LLC
x_{17}	Colonial Heights School Board
x_{18}	Cantu Services Inc
x_{19}	Delhaize America Distribution Center
x_{20}	US Department of Justice
x_{21}	Virginia Department of Transportation
x_{22}	Amazon Fulfillment Service Inc.
x_{23}	City of Hopewell
x_{24}	Perdue Products
x_{25}	Bon Secours Southside Regional Medical Center
x_{26}	Gerdau
x_{27}	Greensville County School Board
x_{28}	DuPont Specialty Products
x_{29}	Amstead Rail Company
x_{30}	Sussex I Correctional Center
x_i	Others

Table 7-16. Case two: criteria-initiative assessment. The criteria-initiative assessment for case two: CPDC employers. Ratings are represented by a filled circle (●, strongly agree), a half-filled circle (◐, agree), an empty circle (○, somewhat agree), and a hyphen (—, neutral)

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
x.01 - US Department of Defense	●	◐	●	●	◐	◐	◐
x.02 - Wal-Mart	◐	○	●	●	—	○	○
x.03 - County of Prince George	●	○	◐	●	—	◐	●
x.04 - Dominion Energy	◐	●	○	●	◐	●	◐
x.05 - Central State Hospital	○	○	—	◐	●	◐	●
x.06 - Boar's Head Provisions Co.	○	—	●	○	—	◐	○
x.07 - Integrity Staffing Solutions	●	◐	●	●	◐		○
x.08 - Greensville Correctional Center	○	—	◐	○	◐	○	●
x.09 - AdvanSix, Inc.	◐	●	●	◐	○	◐	◐
x.10 - City of Petersburg	●	◐	●	●	—	◐	●
x.11 - City of Petersburg School Board	●	○	—	○	—	●	●
x.12 - Hopewell City School Board	●	○	—	○	—	●	●
x.13 - Dinwiddie County School Board	●	○	—	○	—	●	●
x.14 - HCA Virginia Health System	●	●	—	○	—	●	●
x.15 - Food Lion	●	○	●	◐	—	●	○
x.16 - Good Neighbor Holdings LLC	—	○	●	◐	○	◐	○
x.17 - Colonial Heights School Board	●	○	—	○	—	●	●
x.18 - Cantu Services Inc	◐	○	●	◐	○	○	○
x.19 - Delhaize America Distribution Center	●	○	●	●	—	●	○
x.20 - US Department of Justice	◐	◐	●	●	—	○	●
x.21 - Virginia Department of Transportation	●	●	○	●	○	○	●
x.22 - Amazon Fulfillment Service Inc.	●	○	●	○	—	—	◐
x.23 - City of Hopewell	●	◐	●	●	—	◐	●
x.24 - Perdue Products	○	○	◐	●	—	○	○
x.25 - Bon Secours Southside Medical Center	●	●	—	●	○	◐	●
x.26 - Gerdau	●	◐	●	◐	●	●	◐
x.27 - Greensville County School Board	●	○	—	○	—	●	●
x.28 - DuPont Specialty Products	○	◐	●	●	○	●	◐
x.29 - Amstead Rail Company	◐	●	●	●	○	●	◐
x.30 - Sussex I Correctional Center	○	—	◐	○	◐	○	●

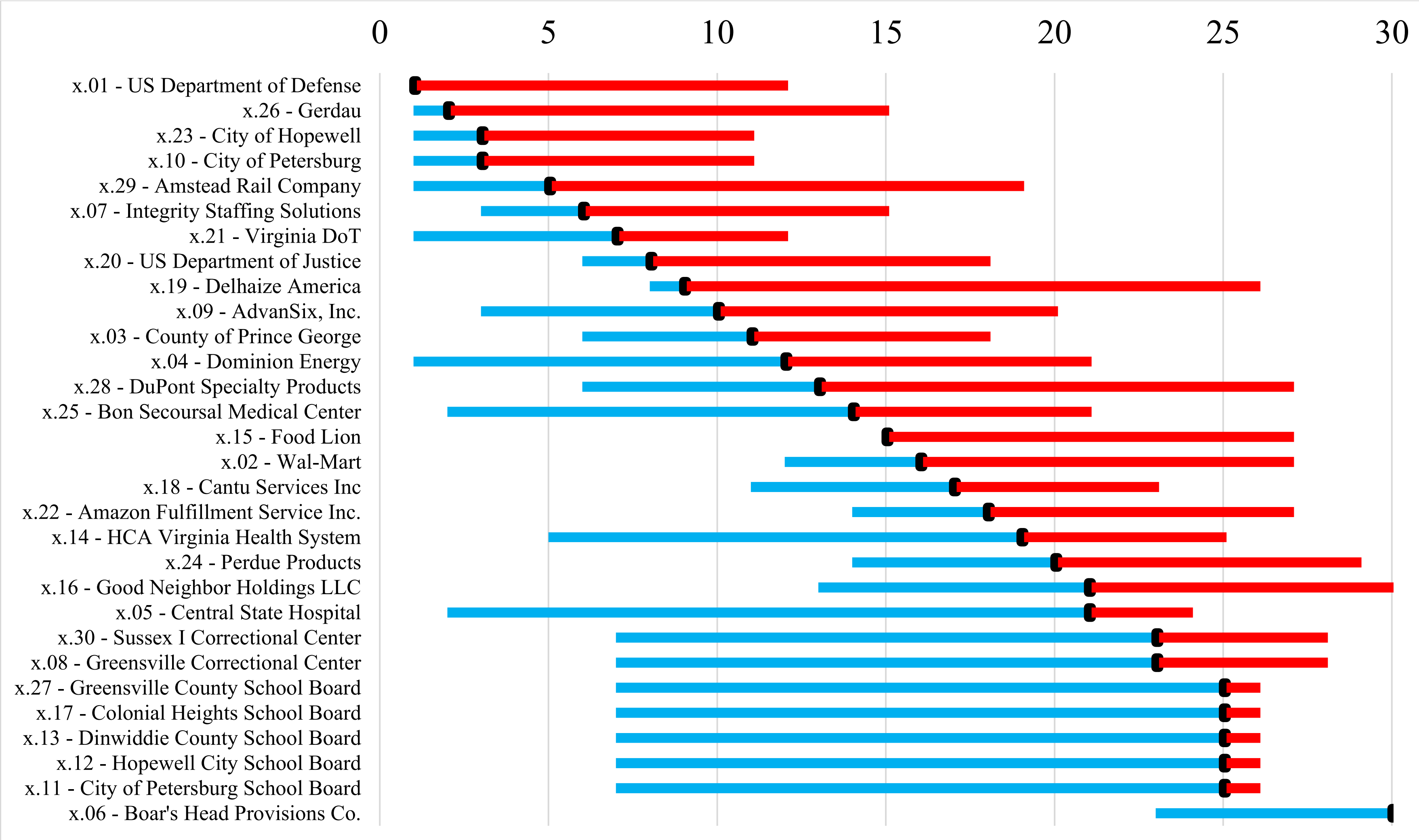

Figure 7-6 Sample of results for case two: CPDC Employers. The results of the analysis of the 30 largest employers of the CPDC.

Table 7-17 Case two initiative ranking chart. This table describes the ranking of each initiative under each scenario for the CPDC employer analysis, the rank matrix $R_{p \times n}$. The green filled cells indicate a higher ranking. The red and orange filled cells indicate a lower ranking.

	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
x.01 - US Department of Defense	1	8	3	12	4	2	2	3
x.02 - Wal-Mart	16	17	27	22	16	15	15	12
x.03 - County of Prince George	11	9	14	6	15	14	18	9
x.04 - Dominion Energy	12	5	4	21	5	1	10	13
x.05 - Central State Hospital	21	16	2	24	21	16	24	20
x.06 - Boar's Head Provisions Co.	30	30	30	29	30	30	23	30
x.07 - Integrity Staffing Solutions	6	11	9	15	7	4	3	4
x.08 - Greensville Correctional Center	23	28	7	25	28	23	18	28
x.09 - AdvanSix, Inc.	10	12	13	20	3	11	5	16
x.10 - City of Petersburg	3	3	11	3	9	9	7	1
x.11 - City of Petersburg School Board	25	21	18	7	23	25	26	23
x.12 - Hopewell City School Board	25	21	18	7	23	25	26	23
x.13 - Dinwiddie County School Board	25	21	18	7	23	25	26	23
x.14 - HCA Virginia Health System	19	15	17	5	13	21	25	22
x.15 - Food Lion	15	19	27	17	16	20	16	17
x.16 - Good Neighbor Holdings LLC	21	26	25	30	22	19	13	19
x.17 - Colonial Heights School Board	25	21	18	7	23	25	26	23
x.18 - Cantu Services Inc	17	20	23	23	18	18	11	18
x.19 - Delhaize America	9	14	26	16	14	13	14	8
x.20 - US Department of Justice	8	7	16	18	11	12	9	6
x.21 - VDOT	7	1	4	1	2	5	12	7
x.22 - Amazon Fulfillment Service Inc.	18	27	24	14	19	22	17	21
x.23 - City of Hopewell	3	3	11	3	9	9	7	1
x.24 - Perdue Products	20	18	29	28	20	17	22	14
x.25 - Bon Secours Medical Center	14	2	6	2	6	6	21	11
x.26 - Gerdau	2	13	1	13	8	7	1	15
x.27 - Greensville County School Board	25	21	18	7	23	25	26	23
x.28 - DuPont Specialty Products	13	10	15	27	12	8	6	10
x.29 - Amstead Rail Company	5	6	10	19	1	3	4	5
x.30 - Sussex I Correctional Center	23	28	7	25	28	23	18	28

The highest priority employer in the baseline scenario is x_1 *– Department of Defense.* This is primarily due to Fort Lee, which employs a large proportion of the population and is responsible for roughly 10% of the economic activity of the region. Further, Fort Lee provides services that are critical to national security, increasing the relative importance of the initiative. x_1 *– Department of Defense* remains ranked highly among all scenarios except *pandemic*, in which it falls to 12th in rank. This is not to say the initiative is less important, but rather the importance of other initiatives supersede x_1 during a *pandemic.* For example, schools and medical clinics all rise in priority in a *pandemic* scenario. x_5 *– Central State Hospital,* x_{14} *- HCA Virginia Health System*, and x_{25} *–Bon Secours Southside Medical Center* rise in relative importance due to the *pandemic* as these facilities must remain open and fully functional to meet the increased demand caused by a *pandemic*.

The second ranked initiative in the baseline scenario, x_{26} *– Gerdau*, is a major manufacturer of the region. This initiative not only produces goods for the region and the nation as a whole but also serves as a recycling center for other manufacturers of the region. The CPDC considered x_{26} *– Gerdau* important in the baseline, but it tends to fall in priority across multiple scenarios, including *pandemic* and *cyber security attack*. x_{19} *- Delhaize America* is a medium sized distribution center for grocery retailers. This is a critical function in the baseline scenario, ranking 9th, however the initiative falls to 26th in the *natural disaster* scenario and tends to fall in priority across other scenarios. This represents one of the largest changes across all employers, and warrants further assessment in subsequent analyses.

Initiatives x_{11}, x_{12}, x_{13}, x_{17}, and x_{27} are all public schools in the CPDC region. In the baseline scenario, all school systems tie as the 25th priority employer. Under the *pandemic* scenario, the schools all rise to 7th in priority. It should be noted that this analysis does not indicate that schools are not important. Rather, the mission of the CPDC to bolster economic development and coordinate projects and policies is not the same mission as governments or the public at large. However, in the *pandemic* scenario when the school systems closed and students began to take classes from home, the importance of the schools rose for the CPDC. Parents staying home with children rather than working during a *pandemic* scenario disrupts the missions of the CPDC as the available workforce decreases. That is, the success criteria chosen by the CPDC are mildly impacted by schools in the baseline (primarily as large employers), but are greatly impacted in the *pandemic* scenario (mostly due to a loss of childcare services). This indicates that the CPDC should dedicate resources to ensuring schools can remain open safely, including promoting vaccinations for students and teachers, maintaining mask use during spikes in infection rates, provide screening for COVID-19 in schools, and recommending social distancing when required (NCIRD 2021).

Figure 7-7 describes the most and least disruptive scenarios to CPDC employers. The most disruptive scenarios for case two, as with case one, are *pandemic* and *natural disaster*. This result is expected – *pandemic* and *natural disaster* both occur abruptly without advanced warning and are destructive. The relative gap in disruptiveness score between *pandemic* and *natural disaster* has widened from 7 to 13 when comparing case one with case two. The CPDC believes this is because of the dual impacts of a *pandemic* scenario. First, *pandemics* cause widespread job loss, wage reduction, and other impacts to employment levels. Second, *pandemics* impact the capacity of governments and industry to deliver goods, services, and amenities to the region. Additionally, uncertainty regarding the length of pandemics and a lack of coordination between the CPDC and its neighboring regions further impact employers. *Funding decrease* is the third most disruptive scenario, though by a wide margin. *Funding decrease* primarily impacts government employers such as x_1 *– Department of Defense*, and x_{21} *– Virginia Department of Transportation.*

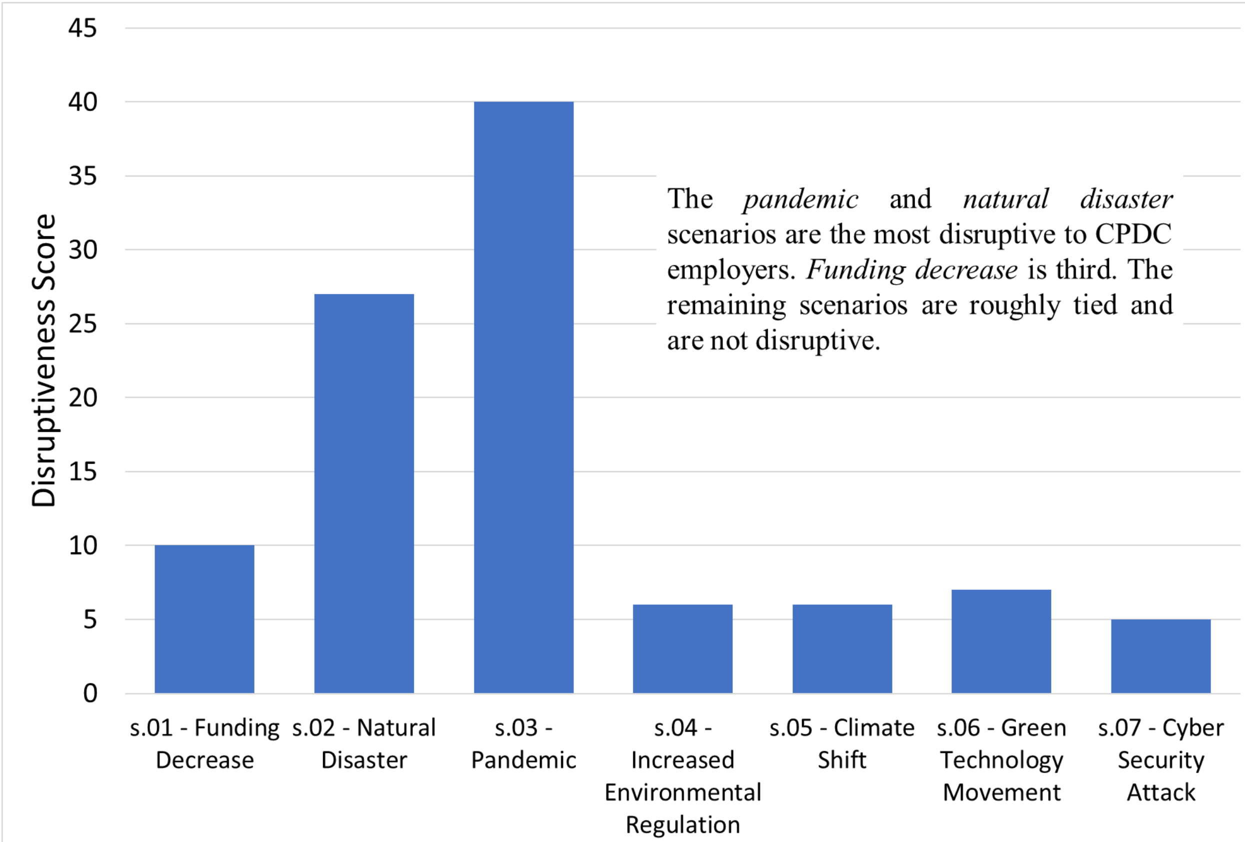

Figure 7-7. Disruptiveness scores of scenarios for CPDC employers. The figure presents the normalized disruptiveness scores across each scenario for the largest employers of the CPDC region. *Pandemics* and *natural disasters* are the most disruptive. The relative disruptiveness of the *pandemic* scenario has risen when compared to case one: CPDC projects.

Table 7-18. Summary of results for case two: CPDC employers

Category	Description
Most disruptive scenarios	s_2 - *Natural Disaster* and s_3 - *Pandemic* are the most disruptive scenarios, with s_3 - *Pandemic* more disruptive.
Least disruptive scenarios	The five remaining scenarios have low disruptiveness scores. However, s_7 - *Cyber Security Attack* is the least disruptive.
Most resilient initiative	The x_1, x_{10}, and x_{23} initiatives are the most resilient. x_1 is ranked the highest the baseline and can fall no farther than rank 12. x_{10} and x_{23} are rank three in the baseline, but can achieve rank one and are never ranked lower than rank 11.
Resilient initiatives	Initiatives x_8, x_{11}, x_{12}, x_{13}, x_{17}, x_{27}, and x_{30} are all resilient initiatives. Though they have low rankings in the baseline, under disruptive conditions they can all rise to rank seven.

7.4.3 Case Three: CPDC Logistics Assets

This section describes the third case for the CPDC regarding logistics assets of the region. Logistics assets such as highways, roads, ports, rail, waterways, utility services, warehouses, distribution centers, and delivery services are all critical elements of the supply chain of the region. Table 7-19 describes the set of initiatives used in case three: CPDC logistics assets. There are 43 assets in total. Assets include 24 commercial and government distribution centers or warehouses, eight critical roads and highways, five sets of utilities, two railroads, and two port assets. The remaining two assets are part of military installations in the region, critical for national distribution of military equipment. Figure 7-20 describes the criteria-initiative assessment for case three.

Table 7-19. The set of initiatives used in case three: CPDC logistics assets. Derived from the Plan RVA report (PlanRVA 2021)

Index	Initiative	Index	Initiative
x_1	Amazon Fulfillment Centers	x_{23}	Moss Motors
x_2	Walmart Distribution Centers	x_{24}	Port Hopewell
x_3	UPS	x_{25}	Norfolk Southern RR
x_4	Aldi	x_{26}	CSX RR
x_5	Perdue Farms	x_{27}	M-64 Marine Highway
x_6	Food Lion Distribution Center	x_{28}	Fort Lee
x_7	Boar's Head	x_{29}	Fort Pickett
x_8	GlaxoSmithKline	x_{30}	Ukrop's Threads
x_9	Defense Supply Center Richmond	x_{31}	I-295
x_{10}	Sabra Dipping LLC	x_{32}	I-95
x_{11}	Reynolds Packaging Group	x_{33}	I-64
x_{12}	PepsiCo	x_{34}	I-85
x_{13}	Maruchan Virginia	x_{35}	Route 58
x_{14}	Medline	x_{36}	Route 10
x_{15}	Campfrio Food Group America	x_{37}	Route 288
x_{16}	Mazda	x_{38}	Route 460
x_{17}	Goya Foods	x_{39}	Internet Services
x_{18}	Ashland Chemical	x_{40}	Waste Management
x_{19}	Church & Dwight	x_{41}	Power Grid
x_{20}	Hill Phoenix	x_{42}	Water Management
x_{21}	Gerdau Ameristeel	x_{43}	Telecommunication
x_{22}	Emerson Ecologics	x_i	Others

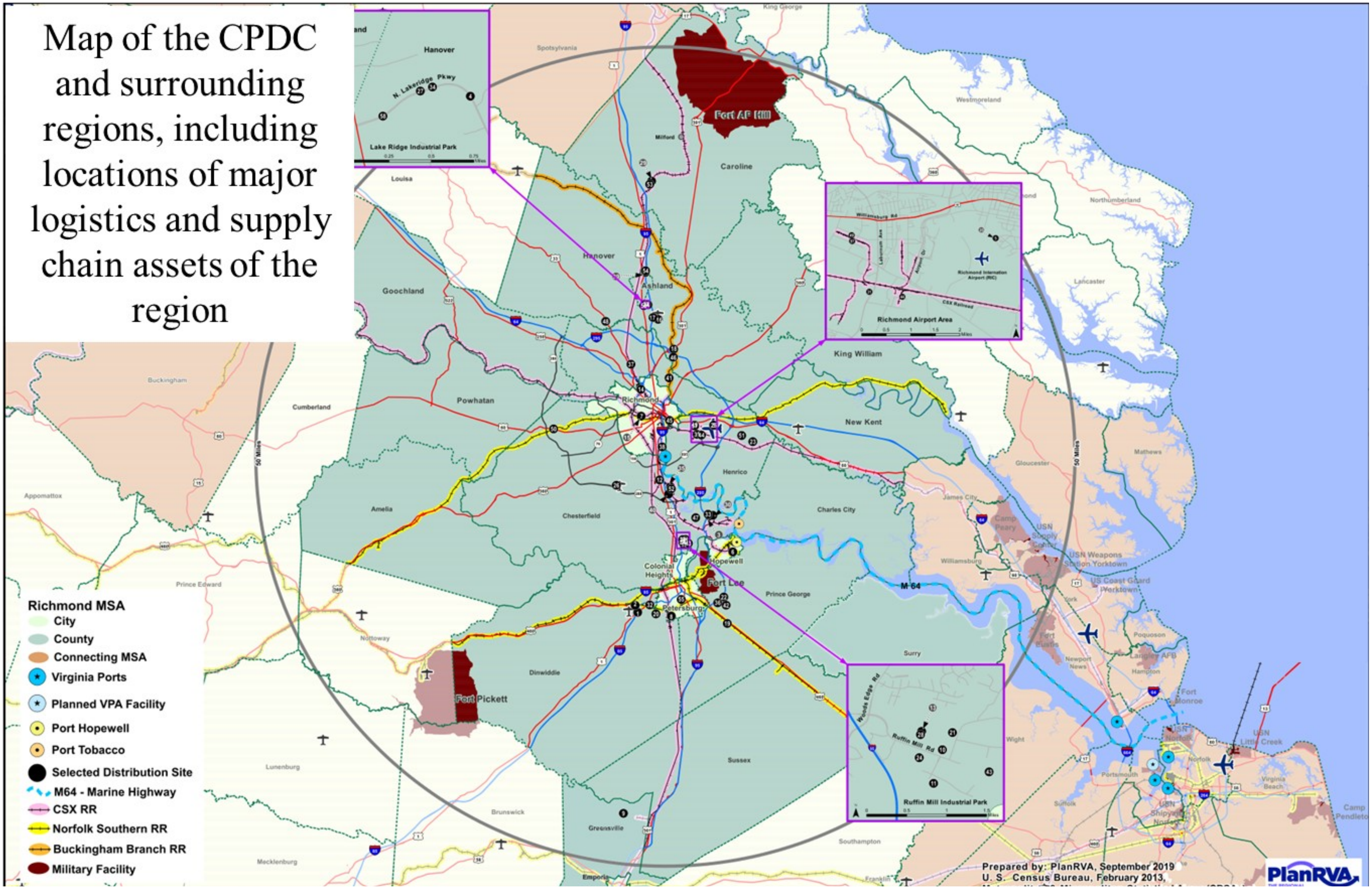

Figure 7-8. Map of the logistics assets of the CPDC Region. This figure shows a map of the CPDC region, marking the locations of critical logistics assets. For case three, some assets that fall just outside of the CPDC region but are nonetheless critical to supply chains of the region are included in the analysis.

Table 7-20. Case three criteria-initiative assessment. The unique criteria-initiative assessment for case three: CPDC logistics assets. Ratings are represented by a filled circle (●, strongly agree), a half-filled circle (◐, agree), an empty circle (○, somewhat agree), and a hyphen (—, neutral)

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
x.01 - Amazon Fulfillment Centers	◐	○	◐	◐	○	○	○
x.02 - Walmart Distribution Centers	●	○	◐	●	○	○	○
x.03 - UPS	●	◐	○	◐	◐	○	○
x.04 - Aldi	●	○	○	○	—	◐	—
x.05 - Perdue Farms	◐	—	○	◐	—	○	—
x.06 - Food Lion Distribution Center	●	—	○	◐	—	○	○
x.07 - Boar's Head	◐	○	◐	●	○	—	—
x.08 - GlaxoSmithKline	○	—	—	○	○	○	◐
x.09 - USDC Richmond	●	◐	◐	●	○	◐	—
x.10 - Sabra Dipping LLC	○	—	◐	○	●	◐	—
x.11 - Reynolds Packaging Group	○	—	◐	—	○	○	○
x.12 - PepsiCo	○	—	○	○	—	○	—
x.13 - Maruchan Virginia	◐	○	○	◐	◐	○	—
x.14 - Medline	◐	○	○	—	○	○	◐
x.15 - Campfrio Food Group America	◐	○	○	◐	—	○	—
x.16 - Mazda	○	—	○	◐	●	○	—
x.17 - Goya Foods	○	—	○	○	—	—	○
x.18 - Ashland Chemical	◐	○	○	—	○	○	◐
x.19 - Church & Dwight	◐	◐	○	◐	○	—	—
x.20 - Hill Phoenix	◐	◐	○	○	◐	◐	—
x.21 - Gerdau Ameristeel	●	◐	◐	○	●	●	—
x.22 - Emerson Ecologics	○	○	—	○	◐	—	○
x.23 - Moss Motors	○	—	○	◐	●	○	—
x.24 - Port Hopewell	○	◐	◐	●	○	○	—
x.25 - Norfolk Southern RR	○	○	◐	●	○	◐	◐
x.26 - CSX RR	○	○	○	◐	○	○	◐
x.27 - M-64 Marine Highway	○	○	◐	●	○	○	○
x.28 - Fort Lee	◐	○	●	●	—	—	○
x.29 - Fort Pickett	○	○	○	◐	—	—	○
x.30 - Ukrop's Threads	○	—	○	●	○	○	—
x.31 - I-296	◐	—	◐	○	○	◐	◐

x.32 - I-95	◐	—	●	●	○	◐	◐
x.33 - I-64	○	—	○	○	—	○	○
x.34 - I-85	◐	—	◐	○	◐	◐	○
x.35 - Route 58	○	—	○	○	◐	○	—
x.36 - Route 10	○	—	○	◐	○	○	—
x.37 - Route 288	◐	—	◐	◐	○	○	—
x.38 - Route 460	◐	—	○	◐	◐	○	—
x.39 - Internet Services	●	◐	●	◐	○	○	—
x.40 - Waste Management	●	○	○	◐	○	◐	●
x.41 - Power Grid	●	○	◐	◐	—	○	—
x.42 - Water Management	●	○	○	○	◐	●	◐
x.43 - Telecommunication	●	◐	○	◐	—	○	—

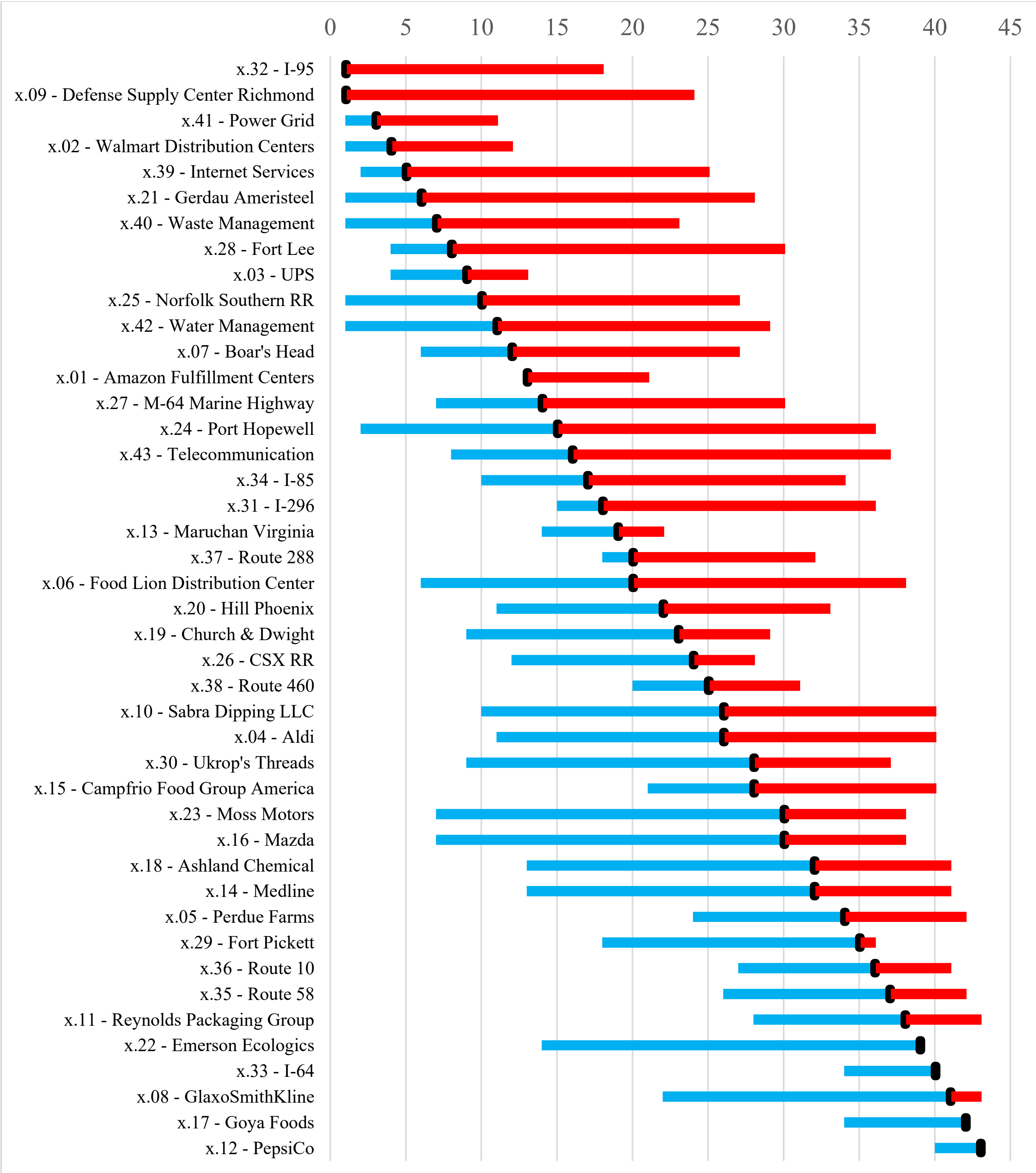

Figure 7-9. Sample of results for case three: CPDC logistics assets. The results of the analysis of 43 critical logistics assets such as roads, rail, distribution centers, and utilities.

Table 7-21. Case three initiative ranking chart. This table describes the ranking of each initiative under each scenario for the CPDC logistic assets analysis, the rank matrix $R_{p \times n}$. The green filled cells indicate a higher ranking. The red and orange filled cells indicate a lower ranking.

	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
x.01 - Amazon Fulfillment Centers	13	14	21	16	15	16	13	14
x.02 - Walmart Distribution Centers	4	2	12	5	7	5	9	1
x.03 - UPS	9	10	5	4	4	4	7	13
x.04 - Aldi	26	35	40	11	22	34	34	32
x.05 - Perdue Farms	34	30	42	26	36	33	39	24
x.06 - Food Lion Distribution Center	20	21	34	6	29	32	38	16
x.07 - Boar's Head	12	11	27	21	14	6	11	6
x.08 - GlaxoSmithKline	41	33	22	29	42	37	43	39
x.09 - DSCR	1	3	24	7	2	1	4	2
x.10 - Sabra Dipping LLC	26	40	11	40	32	28	10	34
x.11 - Reynolds Packaging Group	38	43	31	33	39	43	28	43
x.12 - PepsiCo	43	42	43	43	43	42	42	40
x.13 - Maruchan Virginia	19	20	18	22	19	14	18	19
x.14 - Medline	32	36	16	13	24	38	29	41
x.15 - Campfrio Food Group America	28	22	40	23	21	25	33	21
x.16 - Mazda	30	28	7	38	32	19	20	25
x.17 - Goya Foods	42	38	38	34	40	40	40	35
x.18 - Ashland Chemical	32	36	16	13	24	38	29	41
x.19 - Church & Dwight	23	16	29	19	9	13	18	17
x.20 - Hill Phoenix	22	24	19	20	12	18	11	33
x.21 - Gerdau Ameristeel	6	23	4	9	5	10	1	28
x.22 - Emerson Ecologics	39	31	14	32	26	29	32	37
x.23 - Moss Motors	30	28	7	38	32	19	20	25
x.24 - Port Hopewell	15	8	28	36	6	2	5	8
x.25 - Norfolk Southern RR	10	1	9	27	13	8	14	5
x.26 - CSX RR	24	12	13	28	20	22	27	20
x.27 - M-64 Marine Highway	14	7	23	30	16	9	15	7
x.28 - Fort Lee	8	6	30	17	9	11	8	4
x.29 - Fort Pickett	35	18	36	31	23	27	36	23
x.30 - Ukrop's Threads	28	17	33	37	32	19	34	9
x.31 - I-295	18	26	15	15	27	36	24	30
x.32 - I-95	1	5	6	12	18	12	6	2
x.33 - I-64	40	38	38	34	40	40	40	35
x.34 - I-85	17	34	10	18	27	31	16	31

x.35 - Route 58	37	41	26	42	38	35	31	37
x.36 - Route 10	36	32	35	41	37	30	37	27
x.37 - Route 288	20	27	32	24	29	26	22	18
x.38 - Route 460	25	25	20	25	31	23	26	22
x.39 - Internet Services	5	13	25	8	2	6	2	10
x.40 - Waste Management	7	4	1	1	11	17	23	12
x.41 - Power Grid	3	9	3	3	1	3	3	11
x.42 - Water Management	11	19	1	2	16	24	17	29
x.43 - Telecommunication	16	15	37	10	8	15	25	15

Figure 7-9 describes the results of case three. Initiative x_{32} *- I-95* is the highest priority in the baseline, tied with initiative x_{09} *– US Defense Supply Center Richmond*. x_{32} *- I-95* is a critical interstate highway that connects the CPDC region to neighbors with the nation as a whole. It is critical to the continued operation of the supply chain of the region and is of particular importance to the Port of Virginia. x_{09} *– US Defense Supply Center Richmond* is a hub location for defense supplies to other military installations across the country and internationally. It is important to the region both as an employer, a driver of commerce, and as an important location to the federal government.

There are nine initiatives can each attain ranks one or two. Four utility services (x_{41} *– power grid*, x_{39} *– internet services,* x_{40} *– waste management*, and x_{42} *– water management*) fall into this category, indicating that utility services are highly relevant to the success criteria outlined by the CPDC. These initiatives have the capability to rise in importance under disruption. x_{41} *– power grid* is especially relevant as it does not fall below rank 11 (*cyber security attack*) and remains in the top three across six disruptive scenarios. The other initiatives that can attain ranks one and two are roads, railroads, and the largest distributor and warehouse in the region. However, utilities x_{39}

– *internet services,* x_{40} *– waste management*, and x_{42} *– water management* are volatile and can fall in relative importance due to disruption.

Initiatives x_2, x_4, and x_6 are distribution centers for large grocers in the region. The priority of these initiatives varies in the baseline scenario due to the different sizes and focuses of the distributors, ranking 4th, 26th, and 20th in the baseline respectively. These initiatives all rise in importance under the *pandemic* scenario as access to groceries becomes less certain and more important to the region. Food security is a critical factor in maintaining social cohesion during any disruptive scenario, but especially with rare events such as the COVID-19 pandemic. Contrast the results in the baseline scenario – where the CPDC may prefer to address major roads or railroads – to the *pandemic* scenario where the importance of food security rises. In the *natural disaster* scenario, the relative importance of x_4 and x_6 falls. This is not to say food security is not important during and after a *natural disaster*, but rather that most natural disasters are relatively brief and initiatives that prioritize swift recovery become more relevant. x_2 *– Walmart distribution centers* does not fall due to this effect as the distributor provides many goods beyond groceries that can hasten recovery efforts.

One initiative of particular interest is x_{38} *- Route 460*. This is a major east-west highway that helps the CPDC and the Port of Virginia deliver and receive goods from the maritime port. This roadway runs through the center of the CPDC area and services many of the major assets of the region. Further, x_{38} *- Route 460* intersects with x_{32} *- I-95*, one of the highest ranked initiatives. Stakeholders at the Port of Virginia and within the CPDC noted that x_{38} *- Route 460* appeared to be ranked lower in the baseline than their intuition dictated, and did not respond to disruptive scenarios as they would expect. As such, future analysis may rework performance criteria and criteria-initiative assessments to ensure results align with intuition.

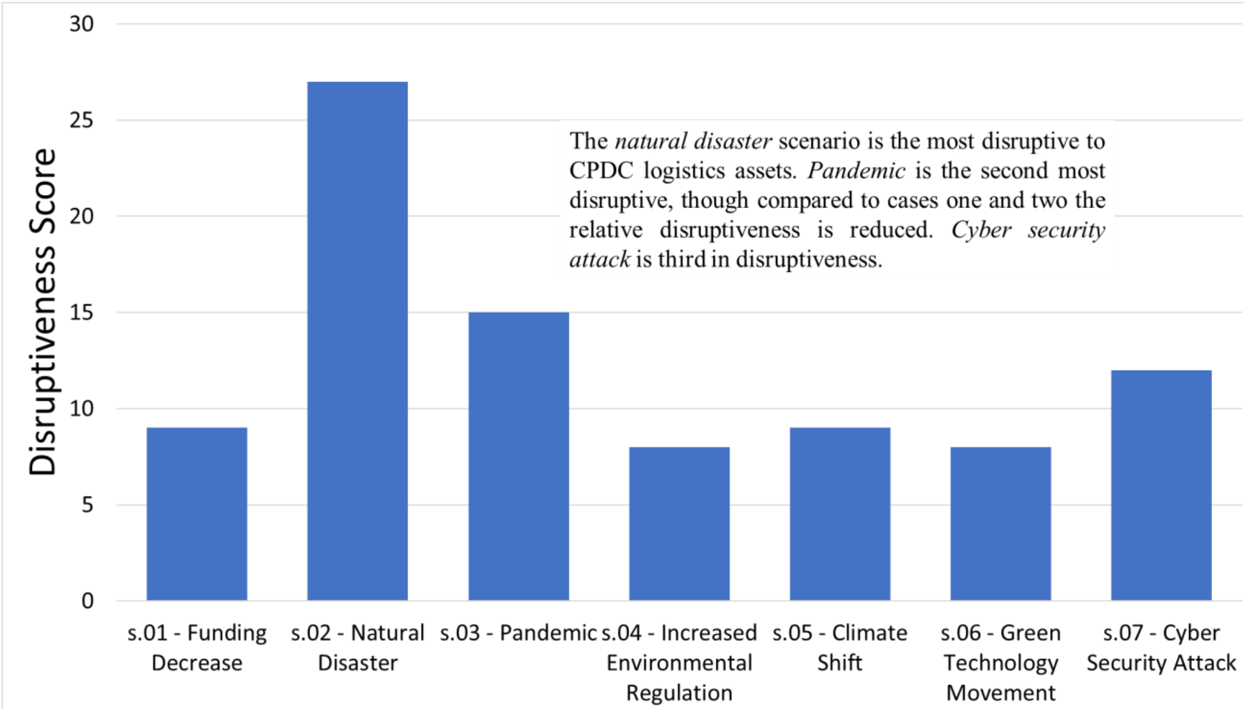

Figure 7-10. Disruptiveness scores of scenarios for CPDC logistics assets. The figure presents the normalized disruptiveness scores across each scenario for the most critical logistics assets of the CPDC region. *Natural disasters* are the most disruptive. The relative disruptiveness of the *pandemic* fallen when compared to the previous two analyses.

Figure 7-10 describes the disruptiveness scores for the logistics assets of the CPDC region. *Natural disaster* is the most disruptive scenario. This matches intuition, as abrupt events such as floods and hurricanes can damage and destroy infrastructure such as roads, warehouses, and rail. The destruction of transportation vectors has a dual impact on the region as the asset itself requires repair and citizens are unable to use these assets in the meantime. Further, *natural disasters* may impact utilities such as the power grid, internet services, and water resources.

When compared to the first two cases, the disruptiveness of *pandemics* has fallen, though it is still the second most disruptive scenario. In *pandemic* scenarios, the primary means of influence is fluctuations in supply and demand due to shortages in goods and in workforce. Staffing shortages may reduce institutional efficacy as there are fewer workers, many workers are undertrained, and external stressors disrupt normal operations.

The third most disruptive scenario is *cyber security attack.* Such attacks are disruptive to individual assets, for example the Colonial Pipeline ransomware attack (Turton and Mehotra 2021). However, the overall disruptiveness of this scenario to the CPDC is limited as the effects are unlikely to cascade throughout the system. That is, if a single organization is attacked the impact is likely siloed only to that organization. Contrast this with *natural disasters* and *pandemics*, which impact multiple organizations at once.

Table 7-22. Summary of results for case three: CPDC logistics assets

Result	Description
Most resilient initiatives	x_{41} - *Power Grid* and x_2 - *Walmart Distribution Center* are two of the most resilient initiatives, both ranking highly in the baseline and remining in the top half of initiatives due to disruption.
Other important initiatives	x_{32} – *highway I-95* and utilities such as x_{39} – *internet services* and x_{40} – *water management* are ranked highly in the baseline, but drop greatly in rank due to disruption.
Most disruptive scenarios	Scenario s_2 – *natural disaster* is the most disruptive scenario, followed by s_3 – *pandemics* and s_7 – *cyber security attack*
Least disruptive scenarios	Scenarios s_1 – *funding decrease,* s_4 – *increased environmental regulation,* s_5 – *climate shift,* s_6 – *green technology movement* are all the least disruptive scenarios

7.4.4 Sensitivity analysis of the mathematical framework

This section provides a sensitivity analysis of the mathematical framework. Specifically, this section investigates how priorities change under different analyst "profiles" by changing the relative importance of success criteria. In decision analysis, much research has focused on the development of stakeholder preference profiles as a way to classify various types of decision makers (Marttunen and Mustajoki 2018). This allows for greater understanding of the varying levels of rationality of decision makers, considering their biases and personalities when analyzing their systems (Kuzmanovic 2019). Developing these profiles allow for new metanalyses to understand how the mathematical framework responds to different types of users (Sadiq *et al.* 2022). In this case the sensitivity analysis takes the form of altering the relative weights of the importance criteria. This sensitivity analysis will focus on an aggressive and a cautious decision maker. Table 7-23 describes the change in weights for an aggressive decision maker. Table 7-24 describes the change in weights for a cautious decision maker.

Table 7-23. Aggressive criterion weights for sensitivity analysis. The updated criterion weights representing an aggressive decision maker. An aggressive decision-maker will give more importance to highly rated criteria compared to a neutral decision-maker. High and medium importance criteria change from 4 to 10 and 2 to 3 respectively.

Criterion Weight	Numerical Weight
High	10
Medium	3
Low	1

An aggressive decision maker is likely to give more weight to the higher priority success criteria (Marttunen and Mustajoki 2018). Initiatives that impact the *high* rated success criteria will be more likely to rise in priority compared to initiatives that impact *medium* or *low* weighted criteria. Figure 7-11 show the change in prioritization from the baseline scenario for an aggressive decision maker.

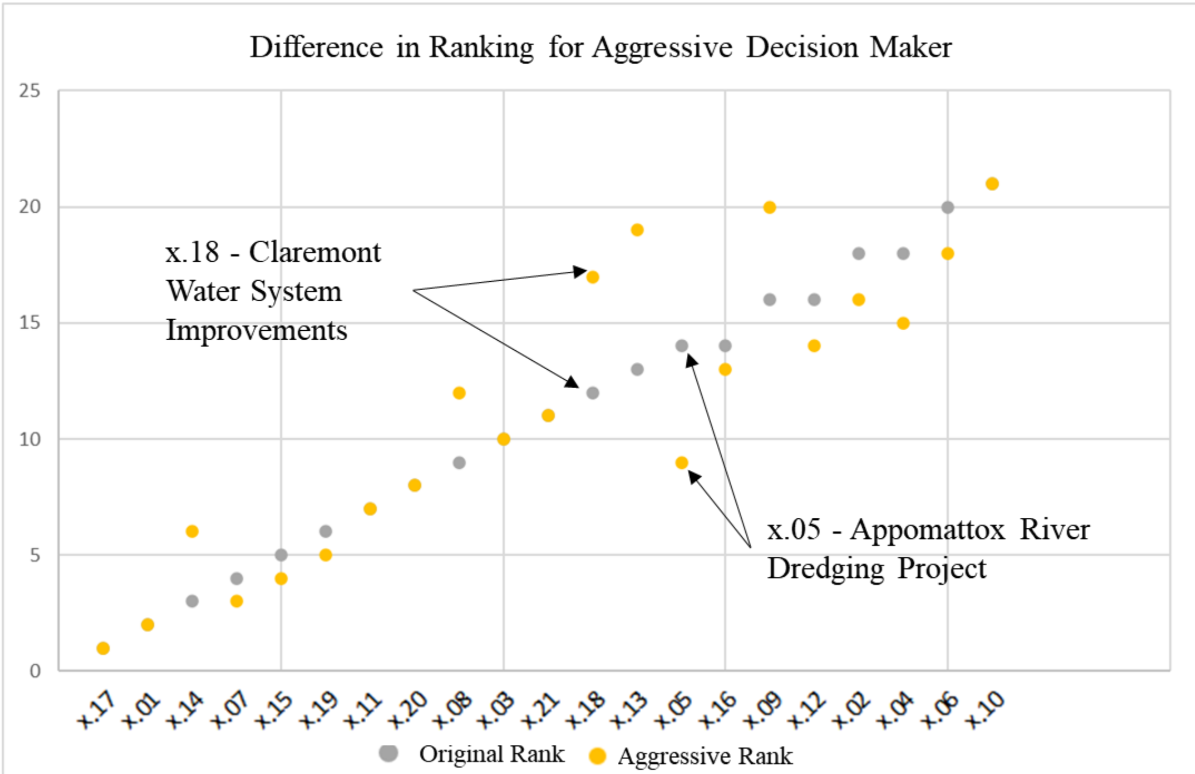

Figure 7-11. Change in priority of CPDC projects for an aggressive analyst. In the circumstance that an aggressive decision-maker is utilizing the framework, many of the mid and low-ranked initiatives in the baseline change in rank. *x.18 – Claremont Water System Improvements* falls in importance while *x.05 - Appomattox River Dredging Project* rises.

In the sensitivity analysis for an aggressive decision maker, there is little change in the top 10 highest rated priorities. The third ranked initiative (x_{14} - *Dendron Area Water System Replacement*) falls to sixth as priorities four through six all rise one position each. Most of the changes occur outside of the top 10 initiatives. For example, *x.18 – Claremont Water System Improvements* falls from 12th to 17th rank, while x_{13} - *Water Line Extension - Mega Site* falls from 13th to 19th. Still others rise in rank for an aggressive decision maker; *x.05 - Appomattox River Dredging Project* rises from 14th to 9th in priority. Broadly, CPDC projects that focused on local economic developments such as site development or improvements to local roads rose in priority, while projects that impacted regional utilities fell in priority. Using the same calculation as Equation 7-3, an analyst can calculate that the disruptiveness of aggressive decision maker when compared to the standard. The aggressive decision weights have a disruptiveness score of roughly six.

A cautious decision maker is likely to give less weight to the high and medium priority success criterion. Compared to an aggressive decision maker, the initiatives with influence on high importance success criterion will be more likely to change in priority. Figure 7-12 shows the change in prioritization in the baseline scenario for a cautious decision maker.

Table 7-24. Cautious criterion weights for sensitivity analysis. The updated criterion weights representing a cautious decision-maker. A cautious decision-maker gives more importance to less rated criteria compared to a neutral decision-maker. High and medium importance criteria change from 4 to 2 and 2 to 1.5 respectively.

Criterion Weight	Numerical Weight
High	2
Medium	1.5
Low	1

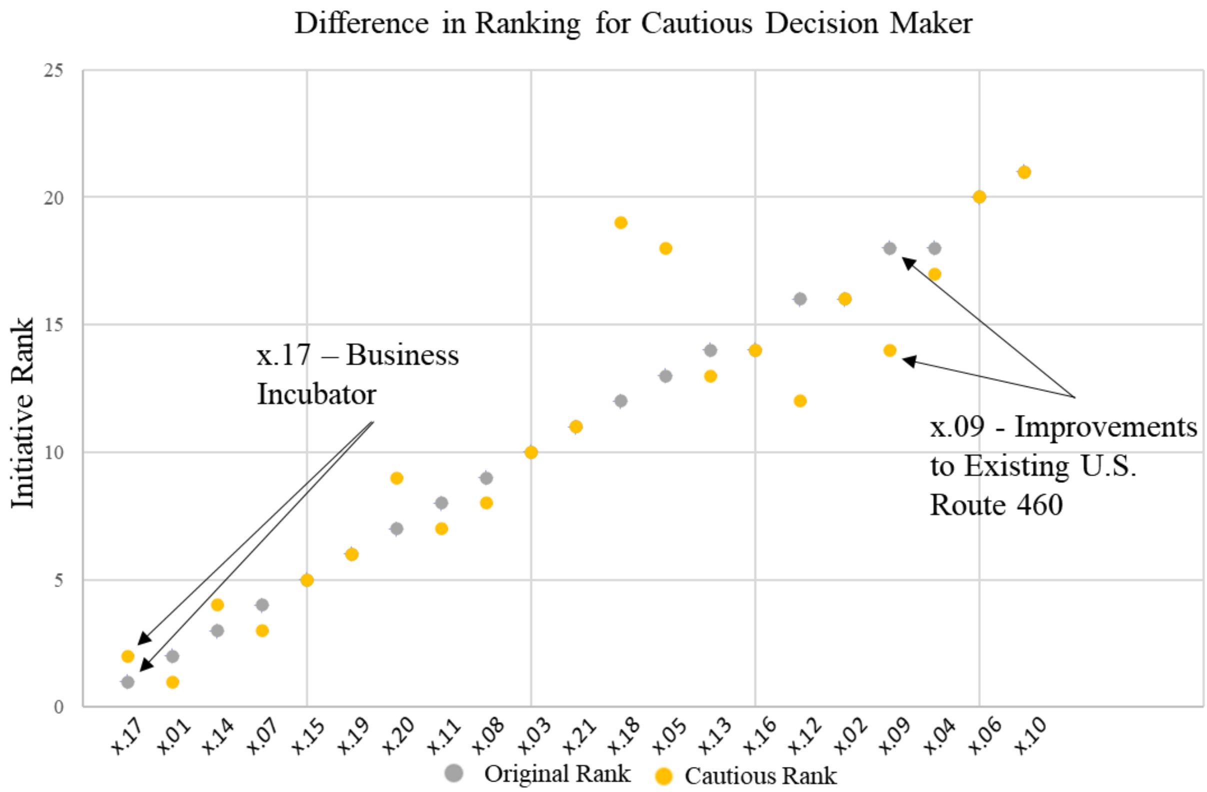

Figure 7-12. Change in priority of CPDC projects for a cautious analyst. In the circumstance that a cautious decision-maker is utilizing the framework, ranks change slightly throughout, but most noticeably in the middle and low rankings.

Compared to an aggressive decision maker, there is greater movement in the top 10 initiatives. In the aggressive case, five of the top 10 initiatives move at least one position. In the cautious case, seven of the top 10 move at least one position, including the top two initiatives. x_{17} *- Business Incubator Collocated w/ Southside Virginia Educational Center*, the top-rated initiative in the baseline switches positions with x_{01} *- MAMaC 1,600 Acre Mega Site*. In both the aggressive and cautious case, *x.18 - Claremont Water System Improvements* and x_{13} *- Water Line Extension - Mega Site* both fall in priority, the same result as the aggressive case. In fact, the results outside of the top 10 initiatives are fairly similar across both analyses. The disruptiveness score of an aggressive decision maker is roughly five. Figure 7-13 shows the relative disruptiveness scores of the user profiles on CPDC priorities.

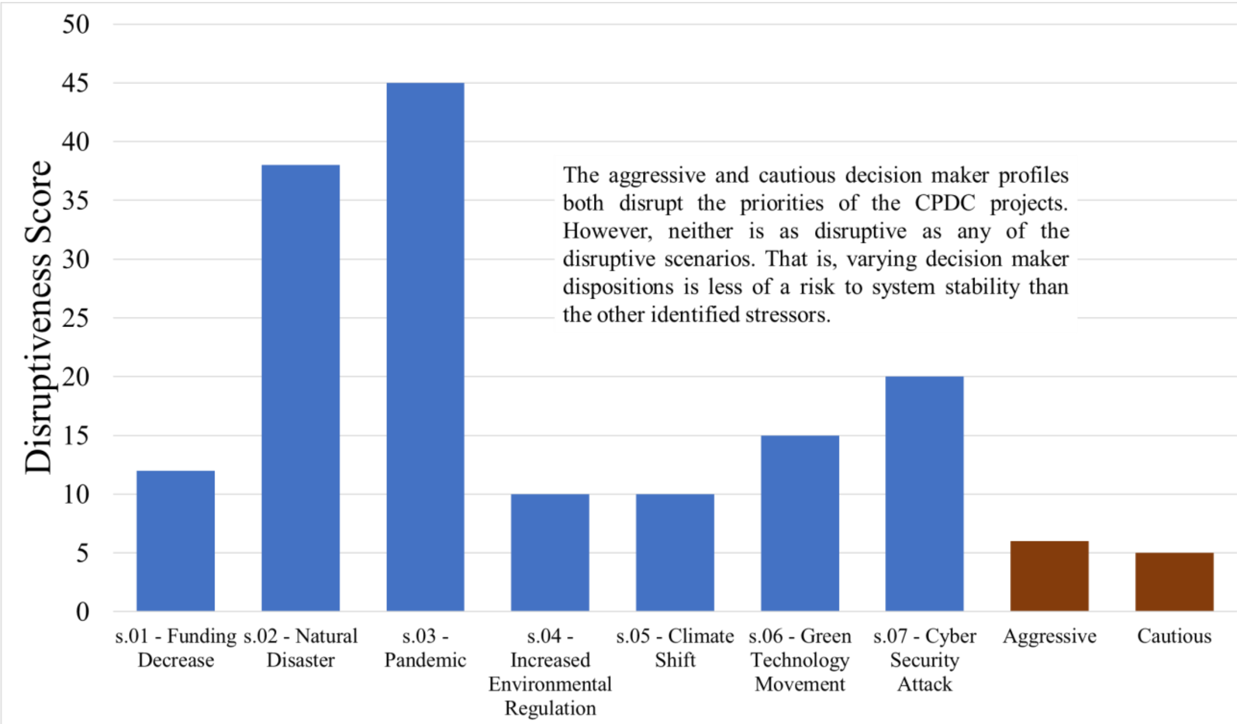

Figure 7-13. The disruptiveness scores for case one: CPDC projects, including the influence of stakeholder disposition on priorities.

7.5 Conclusions from mathematical framework

This section outlines how the results of the framework were used by the CPDC. As this is the final step of the CPSRM, the blue team uses the results of the framework to determine which mitigating actions are feasible to add to the system description, improving overall system resilience. The CPDC outlined a set of 11 major findings based on the results of this analysis, and developed 10 activities that could be performed to reduce the impacts of disruption. Table 7-25 describes the 11 findings. Table 7-26 describes the actions the CPDC could take to mitigate the impacts of disruption. Figure 7-14 describes which actions address which findings. The two most disruptive scenarios were *natural disaster* and *pandemic* across each of the three cases, so the findings and actions are broken down by these two scenarios, however many of the actions outlined here address may of the remaining scenarios.

Table 7-25. Finding and conclusions from the mathematical framework. Based on the outcomes of the case studies, the CPDC identified 11 major findings. These findings include identifying specific impacts of the COVID-19 pandemic, the increased importance of roads, and the need for enhanced business development and job training.

Index	Findings
f.01	Adaptations to Social Distancing
f.02	Disruption to Physical Infrastructure
f.03	Resilient Roadway Improvements
f.04	High Business Development Ranks
f.05	Dual Impact of Pandemics
f.06	Supply Chain Employer Resilience
f.07	Public School Resilience
f.08	High Ranking Utilities
f.09	Critical Roadways
f.10	Avoidance of Downtime
f.11	Disruptions to Military
f.i	Others

Table 7-26. Resilience and mitigating actions. Based on the results of the analysis, the CPDC outlined ten actions that would improve system resilience. These include new projects, employment opportunities, policies, and asset development

Index	Resilience/Mitigating Actions
r.01	Update Development Plans
r.02	Enhance Routes 460 and 58
r.03	Expand Rail Access
r.04	Rural Public Transportation
r.05	Site Development
r.06	Information Sharing
r.07	Employee Training Grants
r.08	On-the-Job Training Grants
r.09	Trucker Training
r.10	Service Industry Investment
r.i	Others

		Resilience/Mitigating Actions									
		Other Stressors					Pandemic Stressors				
		r.01 Update Development Plans	r.02 Enhance Routes 460 and 58	r.03 Expand Rail Access	r.04 Rural Public Transportation	r.05 Site Development	r.06 Information Sharing	r.07 Employee Training Grants	r.08 On-the-Job Training Grants	r.09 Trucker Training	r.10 Service Industry Investment
Case 1: CPDC Projects	f.01 Adaptations to Social Distancing				✓		✓			✓	✓
	f.02 Disruption to Physical Infrastructure	✓	✓	✓		✓				✓	
	f.03 Resilient Roadway Improvements		✓	✓						✓	
	f.04 High Business Development Ranks				✓			✓	✓		✓
Case 2: CPDC Employers	f.05 Dual Impact of Pandemics				✓		✓	✓	✓		✓
	f.06 Supply Chain Employer Resilience	✓	✓	✓				✓		✓	
	f.07 Public School Resilience				✓		✓				✓
Case 3: CPDC Logistics Assets	f.08 High Ranking Utilities	✓	✓		✓	✓	✓	✓	✓		
	f.09 Critical Roadways		✓	✓	✓	✓				✓	
	f.10 Avoidance of Downtime	✓	✓	✓	✓			✓	✓		✓
	f.11 Disruptions to Military	✓			✓	✓	✓				

Figure 7-14. Findings-actions matrix. The matrix used to match which mitigating actions address each finding. These combined results were used to outline the specific plans the CPDC intends to integrate to their system.

The first resilience or mitigating action, r_1 – *update development plans*, is one method for reducing the impact of natural disasters. The activity includes an exploration of the existing urban, natural resource, and land planning policies of the region. Updates to these policies might include an assessment of flood-prone areas, limiting the development and urbanization of repetitive loss areas (Bonato *et al.* 2021). This would reduce the impacts of some natural disasters that include flooding, enabling faster recovery.

r_2 – *enhance routes 460 and 58* is a mitigating action as it directly impacts the Port of Virginia and the major employment sectors of the CPDC. Enhancements would ensure that these roadways are more easily accessible, more reliable with respect to travel times, and reduce stress on supply chains. This would impact the local distribution centers and port assets directly. Changes may include the expansion of the roads, enhancements to roadway geometry, and an increase in access to the roads. Ensuring a high level of service, especially under disruptive scenarios in which roads may be unsafe or failing, can reduce the costs of recovery and improve system resilience (Pennetti et *al*. 2020).

Access to diverse transportation vectors can also reduce the influence of disruptive scenarios. r_3 – *expand rail access* is another activity that improves regional resilience. The development of a rail hub would reduce dependence on roads for the delivery of goods while improving the efficiency of the Port of Virginia operations (Hendrickson 2021). First, if a natural disaster were to damage or otherwise render some roads unusable for an extended period, rail may be a viable alternative for the delivery of goods and for the hastening of recovery efforts. Further, rail can be quickly and easily repaired in the event of a natural disaster, as rapid response repair teams can be deployed to quickly address any damage to rail. The CPDC would partner with the Virginia Department of Rail and Public Transportation as a way to further develop rail to reduce dependence on trucks.

r_4 – *rural public transportation* is another activity that can reduce the impacts of natural disasters and pandemics. The program would provide transit for citizens from rural areas to major urban areas and back. This includes inter-city travel. This service allows citizens to travel conveniently and improve freedom of mobility for residents in the CPDC area. In the event of a natural disaster, reliable public transportation improved recovery times and reduces the harm to affected communities (FCRTA 2021). This service could take many forms – as fleets of electric vehicles, enhanced bussing services, or on demand and point-to-point services for residents without access to private vehicles.

r_5 – site development is another method for increasing resilience to natural disasters for the CPDC. Site development is the process of readying land for new building development. Land is graded in five tiers. A tier one site has very little or no development, while a tier five site is considered "shovel ready", and construction could begin immediately (VEDP 2021). Preparing sites – that is, creating more land that is of high-tier – is a method for attracting businesses. This would drive economic development of the region, and when accomplished with careful planning with respect to natural disasters, enables the CPDC to more quickly recover from natural disasters while reducing the initial impact.

r_6 – information sharing, is an activity specific to *pandemics*. One of the major contributing factors to prolonging the COVID-19 pandemic in the CPDC region was a lack of coordination among neighboring localities. Differing rules and regulations led to confusion, slowing the reopening of many businesses (NCIRD 2021). Developing an information sharing plan is one way the CPDC could ensure all of their constituent members coordinate their response to pandemics. This would include shared dashboards, data, and policies between localities.

r_7 – employee training grants and *r_8 – on-the-job training grants* are investments to increase the skill of the workforce. Employee training grants are given to individuals to fund their training an enhance their skills. This would include funds for getting new certifications and general skill improvements such as forklift certifications or computer programming. On-the-job training grants allow an employer to hire underqualified candidate to a position with the knowledge that

they will be compensated for training the individual. The grants are also used to advertise the job openings, fund job fairs, and increase outreach.

r_9 – *trucker training* is of particular interest to the Port of Virginia and of high relevance to the demonstration of the CPSRM presented in this dissertation. Highly skilled truckers can reduce bottlenecks in port processes, particularly the container stacking problem. Further, a high volume of available truckers reduces stress on the port during disruptive scenarios. This would be especially effective in *pandemic* and *natural disaster* scenarios.

r_{10} – *service industry investment* is direct assistance for businesses that are most vulnerable to pandemics. Hospitality and food service industries were the most severely impacted by COVID-19 and were the slowest to recover in the CPDC region (Chmura 2021). This would take the form of grants or other funding to reduce stress on the service industry.

With the framework completed, the blue team will determine which resilience capabilities should be funded and applied to the system. The previous chapters feed into this assessment. Though the Port of Virginia performed the previous stages of the CPSRM for the demonstration in this dissertation, data constraints led to a shift for the final stage in this chapter focusing on a major sociotechnical system within the region of influence of the Port, the CPDC. However, the models and resilient solutions developed in the previous chapters would be included here, in the set of final feasible recommendations based on findings. That is, the one of the findings is f_{10} – *avoidance of downtime*. This finding can be resolved by a new resilience/mitigating action for the port, the implementation of a reinforcement learning algorithm for controlling container stacking processes. All stages of the CPSRM have now been completed. The CPSRM can be executed several times as system requirements and objectives change.

Chapter 8: Summary and conclusion

8.1 Overview

Section 8.2 reviews the content of the dissertation. Section 8.3 describes future work and research directions based on the dissertation.

8.2 Summary

This dissertation includes six contributions to the theory, methodology, and execution of the systems engineering discipline. These contributions are dissimilated across 10 papers or published reports and four presentations. Figure 8-1 shows the relationship between the methodology presented in this dissertation and chapters. Table 8-1 describes the relationship between chapters, publications, and contributions.

Chapter 3 presented the CPSRM, a methodology for the design and analysis of cyber-physical systems. The CPSRM aligns with best practices for CPS design, analysis, and requirements elicitation. The CPSRM advances on previous work inf CPS analysis and design by 1) incorporating elements from other disciplines including cybersecurity and risk analysis, and 2) incorporating multidisciplinary teams across all design phases. The CPSRM was applied to the Port of Virginia and the surrounding region. The outcome of the analysis is a set of new design requirements for increasing resilience to disruption.

Chapter 4 presented a system specification, hazard analysis, and gap analysis and represents stages one and two of the CPSRM. The system specification utilizes work from computer science to elicit design requirements for CPS, determining the missions, goals, needs, and requirements of the system to be used in a system simulation. The chapter outlines the container management process of the Port of Virginia, explaining the requirements for developing a simulation. The hazard analysis provides additional design considerations, outlining the impacts of disruptive events on operations. The gap analysis provides a set of techniques and methods that can be applied to the system to improve resilience to disruption.

Chapter 5 presented a simulation of the container handling process in the Port of Virginia, and represents stages three, four, and five of the CPSRM. The simulation outlines the environment, rules, and dynamics of a container stacking block – a storage location for containers awaiting pickup at the port. The simulation is highly customizable, allowing users to alter the size and shape of the container stacking block, the number and rate of incoming containers, the dwell time of containers, and many other factors. The simulation is the test bed for the MuZero reinforcement learning algorithm used to manage the container stacking environment and serve as a resilience technique for resisting the impacts of disruption.

Chapter 6 presented the implementation of the MuZero reinforcement learning algorithm as a controller for the container stacking block, and represents stages three, four, and five. The algorithm does not require data to train, instead learning system dynamics using the simulator. MuZero is able to determine which container moves are best given the current state of the container stacking block environment. The algorithm achieved results similar to the Port of Virginia in the baseline case on an example environment. The algorithm was also used to show how different stacking block configurations change performance. MuZero was used to explore how disruptive scenarios impact operations, and was able to reduce the impacts of these disruptions.

Chapter 7 presented a mathematical framework for analyzing the impacts of disruptions to system priorities and represents stage six of the CPSRM. This work advances the mathematical framework by introducing analyst profiles, exploring how different levels of risk tolerance influence the disruption of priorities. Using a region near the Port of Virginia, the framework is used to explore how disruptions influence system priorities across projects, employers, and logistics assets of the region. This information is used to develop the final system description in the CPSRM with new resilience requirements.

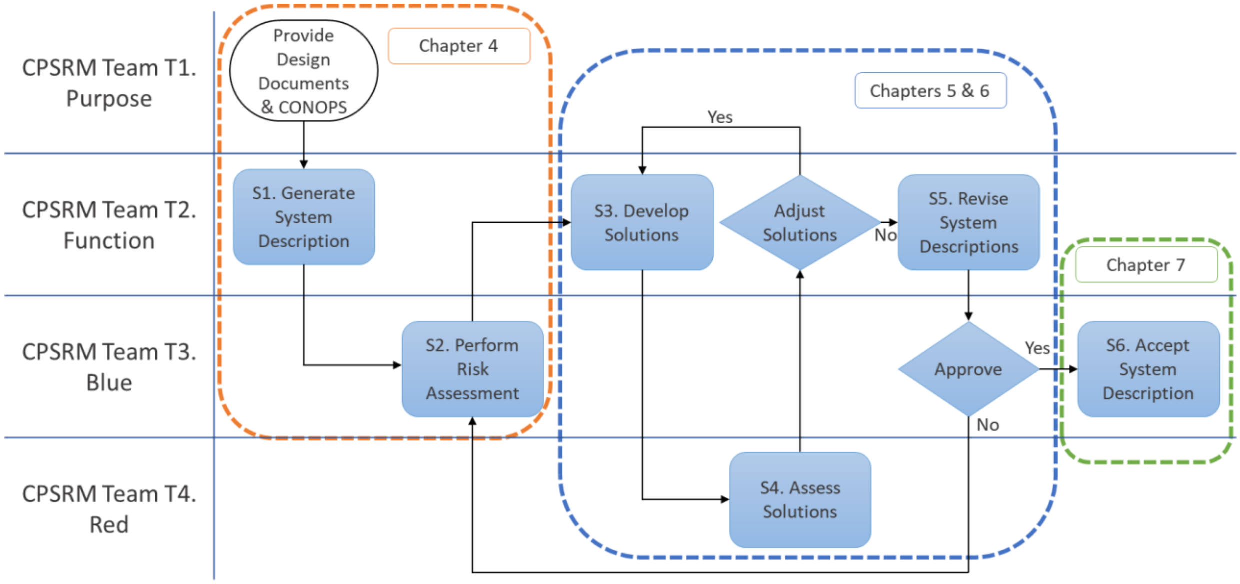

Figure 8-1. Overview of CPSRM and the relationship of each stage to chapters

Table 8-1. Summary of relationship between chapters, presentations or publications, and research contributions

Chapter	Presentation or Publication	Contribution
Ch.3, Ch.5	SERC, Beling, Loose *et al.* 2020	I, II, III, IV
Ch.3, Ch.5	SERC, Beling, Loose *et al.* 2021	I, II, III, IV
Ch.7	CESUN, Loose *et al.* 2021	VI
Ch.7	CPDC, Lambert, Loose *et al.* 2022	VI
Ch.7	CPDC, Loose *et al*. 2022(a)	VI
Ch.7	IEEE SYSCON, Loose *et al.* 2022(b)	VI
Ch.3, Ch.7	IEEE SYSCON, Eddy, Loose *et al.* 2022	I, VI
Ch.7	IEEE SYSCON, Loose *et al.* 2022(c)	VI
Ch.5, Ch.6	IEEE SIEDS, Hamdy, Loose *et al.* 2022	IV, V, VI
Ch.7	Wiley journal *Risk Analysis*, Andrews, Loose *et al.* 2022	VI
Ch.7	Submitted Wiley journal *Risk Analysis*, Loose *et al.* 2023(a)	VI
Ch.5	Wiley journal *Systems Engineering*, Eddy, Loose *et al.* 2023	II, VI
Ch.4	VDEM HIRA, VDEM, Loose *et al.* 2023	II, III
Ch.5, Ch.6	IEEE SIEDS, Costello, Loose *et al*. 2023	III
Ch.4, Ch.7	Working Paper, Hill, Loose *et al.* 2023	III
Ch.5, Ch.6	Submitted IEEE CoDIT, Loose *et al.* 2023(b)	IV, V

8.3 Future work

Each contribution developed in this dissertation is critical to the continuous improvement of systems engineering and resilient design. Each method can be expanded and refined to apply to a broader group of users or stakeholders. The CPSRM itself can be expanded to apply to other application domains outsaid of maritime logistics. For example, transportation planning has several relevant research areas such as ramp metering. These types of systems are excellent future use cases as they have cyber-physical components, a need for robust resilience measures, and high-fidelity simulation models.

The system specification process can be expanded to incorporate “fuzzy requirements”. Traditional requirements have a defined, input, output, and function – fuzzy requirements allow stakeholders to outline less specific criteria. For example, a fuzzy requirement may be “the cost shall be low” without a specific constraint on budget. This type of requirement allows T2 greater freedom when developing a system description and specification.

The hazard analysis process may also evolve to focus on the outcomes of disruptive events rather than the inciting incident. Many hazards have similar impacts on systems, and can be protected by the same resilience technique. For example, a snowstorm and flood have similar impacts on transportation systems – roads may close and utilities may be disrupted. Creating diverse and accessible transportation vectors utilizing a variety of fuel sources can reduce the impact of these disruptions. Understanding the impact of disruptions will help allocate resources in the final systems design

The simulation model of the port operations can be expanded and refined to include more realistic parameters. For example, the current model focuses on container moves only. However, the port also prioritizes minimizing truck turn time – the amount of time a truck is on the terminal. Additionally, touches per container are used as a proxy for energy use – however, energy may be recovered as containers descend, increasing the reward for stacking containers to high tiers. The port also has lower energy costs at night – moving containers when the terminal is not in operation is another cost saving measure. Future explorations of this simulation will explore these additional measures to improve simulation fidelity.

The reinforcement learning algorithm used as the process controller can also be expanded and generalized. The MuZero algorithm can be expanded to include stochastic processes, which is necessary to integrate feedback such as variable energy prices. It would benefit the machine learning community to establish a standard test bed for the container handling process so new stacking techniques can be tested against one another. This future effort should establish consistent performance metrics and environment sizes. Further, future work will adjust the current model parameters to apply to environments of any size. The current model is limited to action parameters that it is trained on – however, a general-purpose reinforcement learning algorithm that can make decisions in any environment of any size, making the model much more flexible and useful to stakeholders.

References

Abduljabbar, R., Dia, H., Liyanage, S. and Bagloee, S.A. "Applications of artificial intelligence in transport: An overview." *Sustainability*, *11*(1), p.189. 2019

Aegis Environmental. "Comprehensive air emissions inventory 2017 update". The Port of Virginia. November 2019.

Ali, S, Ronaldson, S. "Ordinal preference elicitation methods in health economics and health services research: using discrete choice experiments and ranking methods." *British Medical Bulletin*. 2012 ;103(1):21-44.

Almutairi, A. "Risk-based planning and scheduling: a simulation approach to vessel scheduling." Presentation to Port of Virginia Stakeholders. Port of Virginia, Norfolk, Virginia. February 2016.

Almutairi, A. "SNIT improvement strategies and expected impacts of disruptive events at SNIT" Presentation to Port of Virginia Stakeholders. Port of Virginia, Norfolk, Virginia. July 2017.

Almutairi, A., S. AlKheder, S. Alzwayid, D. Talib, M. Ben Heji, and J.H. Lambert. "Scenario-based preferences modeling to investigate port initiatives resilience." *Technological Forecasting and Social Change* 176 (2022): 121498.

Alsultan, M., J. Jun, and J. H. Lambert. "Program evaluation of highway access with innovative risk-cost-benefit analysis." *Reliability Engineering & System Safety* 193 (2020): 106649.

Alsultan, M., Z. A. Collier, and J. H. Lambert. "Risk-based priority setting for large-scale access management programs with uncertain mobility benefits and costs." *ASCE-ASME Journal* *of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 7, no. 1 (2021): 04020046.

Ambrosino, D., and Anna S. “A shipping line stowage-planning procedure in the presence of hazardous containers”. *Maritime Economics & Logistics*. https ://doi.org/10.1057/s41278-018-0107-y. 2018.

American Association of Port Authorities (AAPA). “Not your father’s cranes and equipment”. *AAPA Seaports Newsletter* 2023. Retrieved from https://www.aapaseaports.com/index.php/2019/11/05/not-your-fathers-cranes-and-equipment/

American Society of Civil Engineers (ASCE). “Report card for America’s infrastructure: ports”. *ASCE's 2021 Infrastructure Report Card*. 2022.

Andrade, S. R., and D. E. Hulse. "Evaluation and improvement of system-of-systems resilience in a simulation of wildfire emergency response." *IEEE Systems Journal* (2022).

Andrews, D. J., C. A. Pennetti, Z. A. Collier, T. L. Polmateer, and J. H. Lambert. "Systems evaluation for defense operations of maritime transport." In *2020 IEEE International Systems Conference (SysCon)*, pp. 714-720. IEEE, 2020.

Andrews, D. J., C. A. Pennetti, Z. A. Collier, T. L. Polmateer, and J. H. Lambert. "Segmented identification of disruptive settings on transportation corridors." In *International Conference on Transportation and Development 2020*, pp. 147-158. Reston, VA: American Society of Civil Engineers, 2020.

Andrews, D. J., Eddy, T. L., Hollenback, K. S., Sreekumar, S., Loose, D. C., Pennetti, C. A., ... & Lambert, J. H. “Enterprise risk management for automation in correctional facilities with pandemic and other stressors”. *Risk Analysis.* 2022.

Andrews, D.J., T. L. Polmateer, J. P. Wheeler, D. L. Slutzky, and J. H. Lambert. "Enterprise risk and resilience of electric-vehicle charging infrastructure and the future mobile power grid." *Current Sustainable/Renewable Energy Reports* 7, no. 1 (2020): 9-15.

Bakirtzis, G., Carter, B.T., Elks, C.R., Fleming, C.H. “A model-based approach to security analysis for cyber–physical systems.*” In Proceedings of the 2018 Annual IEEE International Systems Conference (SysCon2018)*, Vancouver, BC, Canada, 23–26 August 2018.

Bakirtzis, G., Sherburne, T., Adams, S., Horowitz, B.M., Beling, P.A. and Fleming, C.H. “An ontological metamodel for cyber-physical system safety, security, and resilience coengineering.” *Software and Systems Modeling*, 2022. 21(1), pp.113-137.

Beling, P. A., B.M. Horowitz, C.H. Fleming, S. A. Adams, T. Sherburne, D. C. Loose. “RT-1013: security engineering. *SERC-2020-TR-008:* Risk based approach to cyber vulnerability assessment.” *2020 Systems Engineering Research Center Research Review, SERC*. 2020.

Beling, P. A., B.M. Horowitz, C.H. Fleming, S. A. Adams, D. C. Loose, T. Sherburne. “Risk based approach to cyber vulnerability assessment using static analysis.” Presentation in *Conference on Systems Engineering Research 2020*. 2020.

Bello, I., Pham, H., Le, Q. V., Norouzi, M., & Bengio, S. “Neural combinatorial optimization with reinforcement learning.” (2016). arXiv preprint arXiv:1611.09940.

Benutzer. “Red shipping container of Lorenzo Shipping Corporation.” Photograph by Benutzer KMJ, distributed under CC BY-SA 3.0. 2004

Beteto, A., V. Melo, J. Lin, M. Alsultan, E. Mario Dias, E. Korte, D. A. Johnson, N. Moghadasi, T. L. Polmateer, and J. H. Lambert. "Anomaly and cyber fraud detection in pipelines and supply chains for liquid fuels." *Environment Systems and Decisions* (2022): 1-19.

Biden, J. “Remarks by President Biden on inflation and actions taken to lower prices and address supply chain challenges”. *White House Press Briefing Room: Speeches and Remarks*. June 2022.

Blum, D., Arroyo, J., Huang, S., Drgoňa, J., Jorissen, F., Walnum, H.T., Chen, Y., Benne, K., Vrabie, D., Wetter, M. and Helsen, L. “Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings.” *Journal of Building Performance Simulation*, *14*(5), (2021) pp.586-610.

Bødker, S. “Through the interface: A human activity approach to user interface design.” *CRC Press*, 2021.

Bonato M., M. Shekarian, and Ma. M. Parast. "The impact of flexibility and redundancy on improving supply chain resilience to disruptions." *International Journal of Production Research.* 2021. 60, no. 6 (2022): 1992-2020.

Bonato, M., Sambo, B., Sperotto, A., Lambert, J. H., Linkov, I., Critto, A., Torresan, S., and Marco, A. “Prioritization of resilience initiatives for climate-related disasters in the metropolitan city of venice,” *Risk Analysis*, 2021

Bosch, R. R. “Solving the dynamic container relocation problem to minimize transportation costs using approximate dynamic programming” (Master's thesis, University of Twente). 2022.

Bouchaut B. and Asveld L. “Safe-by-design: Stakeholders’ perceptions and expectations of how to deal with uncertain risks of emerging biotechnologies in the Netherlands.” *Risk Analysis*. 2020;40(8):1632-1644.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. “OpenAI Gym.” *ArXiv Preprint ArXiv:1606.01540*. (2016).

CalcMaps. “Map radius calculator – 500 mile radius from NIT”. *Calcmaps.com.* Retrieved from https://www.calcmaps.com/map-radius/ . 2015

Carey, J. M., & Rossler, K. “The how when why of high-fidelity simulation.” *Europe PMC*. July 2020.

Carlo, H. J., I. F. A. Vis, and K. J. Roodbergen. 2014. “Storage yard operations in container terminals: literature overview, trends, and research directions”. *European Journal of Operational Research*, 235(2):412–430.

Carter, B., Adams, S., Bakirtzis, G., Sherburne, T., Beling, P., Horowitz, B., & Fleming, C. A preliminary design-phase security methodology for cyber–physical systems. *Systems*, *7*(2), 21. (2019)

Carter, B.T. “Model-based methodology for identifying and evaluating resiliency strategies for cyber-phsyical systems”. Master’s Thesis, University of Virginia, Charlottesville, VA, USA, 2018.

Carvalho, H., A. P. Barroso, V. H. Machado, S. Azevedo, and V. Cruz-Machado. "Supply chain redesign for resilience using simulation." *Computers & Industrial Engineering* 62, no. 1 (2012): 329-341.

Cassottana, B., Roomi, M. M., Mashima, D., & Sansavini, G. “Resilience analysis of cyber-physical systems: A review of models and methods.” (2023). *Risk Analysis*.

Center for Strategic International Studies (CSIS). “Building supply chain resilience.” *Commentary, December 2022 Issue*. December 2022. Retrieved from https://www.csis.org/analysis/building-supply-chain-resilience

Chao, S. and Lin, P. “Minimizing overstowages in master bay plans of large containerships.” *Maritime Economics & Logistics*. 2019. https ://doi.org/10.1057/s4127 8-019-00126 -6.

Chen, S. "A machine learning-based approach to cybersecurity and safety of model predictive control systems." PhD diss., University of California, Los Angeles, 2022.

Chmura, “Economic impact of covid-19 on Crater pdc and localities,” Crater Planning Discrict Commission, Tech. Rep., January 2021.

Cho, S.W., Park, H.J., and Lee, C. “An integrated method for berth allocation and quay crane assignment to allow for the reassignment of vessels to other terminals”. *Maritime Economics & Logistics*. https ://doi.org/10.1057/s4127 8-020-00173 -4. 2021.

Chopra, S., and M. Sodhi. "Reducing the risk of supply chain disruptions." *MIT Sloan management review* 55, no. 3 (2014): 72-80.

Chou, C. and Fang, P. “Applying expert knowledge to containership stowage planning: An empirical study”. *Maritime Economics & Logistics.* 2018. https://doi.org/10.1057/s41278-018-0113-0.

Chowdhury, N., & Gkioulos, V. “A framework for developing tabletop cybersecurity exercises.” In *Computer Security. ESORICS 2022 International Workshops: CyberICPS 2022, SECPRE 2022, SPOSE 2022, CPS4CIP 2022, CDT&SECOMANE 2022, EIS 2022, and SecAssure 2022, Copenhagen, Denmark, September 26–30, 2022, Revised Selected Papers* (pp. 116-133). (2023, February). Cham: Springer International Publishing.

Coad, P., and Yourdon, E. “Object-oriented analysis.” *Yourdon press*, 1991.

Coelho, F., Relvas, S. and Barbosa-Póvoa, A.P., “Simulation-based decision support tool for in-house logistics: the basis for a digital twin.” *Computers & Industrial Engineering*, *153*, p.107094. 2021.

Collier, Z. A., and J. H. Lambert. "Managing obsolescence of embedded hardware and software in secure and trusted systems." *Frontiers of Engineering Management* 7, no. 2 (2020): 172-181.

Collier, Z. A., and J. H. Lambert. "Measurement and tracking of project activity quality based on time and cost indicators." In *ARBS 2021 Proceedings 8th Annual Conference Held Virtually on March 26, 2021 Hosted by Eastern Kentucky University Richmond, KY*, vol. 2, p. 32. 2021.

Collier, Z. A., and J. H. Lambert. "Value of risk information in negotiations with evolving preferences." *Journal of Risk Research* 23, no. 10 (2020): 1353-1369.

Collier, Z. A., J. H. Lambert, and I. Linkov. "Advanced analytics for environmental resilience and a sustainable future." *Environment Systems and Decisions* 41, no. 1 (2021): 1-2.

Collier, Z. A., J. H. Lambert, and I. Linkov. "Algorithms and models for decision making in advanced technology systems." *Environment Systems and Decisions* 41, no. 2 (2021): 179-180.

Collier, Z. A., J. H. Lambert, and I. Linkov. "Decision making to address complexity in systems and organizations." *Environment Systems and Decisions* 41, no. 4 (2021): 485-486.

Collier, Z. A., J. H. Lambert, and I. Linkov. "Integrating data from physical and social science to address emerging societal challenges." *Environment Systems and Decisions* 41, no. 3 (2021): 331-333.

Collier, Z. A., J. H. Lambert, and I. Linkov. "Interdisciplinary mathematical methods for societal decision-making and resilience." *Environment Systems and Decisions* 40, no. 1 (2020): 1-2.

Collier, Z. A., J. H. Lambert, and I.Linkov. "Concurrent threats and disasters: modeling and managing risk and resilience." *Environment systems and decisions* 40, no. 3 (2020): 299-300.

Collier, Z. A., J.H. Lambert, and I. Linkov. "Analytics and decision-making to inform public policy in response to diverse threats." *Environment Systems and Decisions* 40, no. 4 (2020): 463-464.

Collier, Z. A., T. L. Polmateer, and J. H. Lambert. "Risk management decision making for security and trust in hardware supply chains hardware supply chain." *CESUN 2020* (2021).

Collier, Z. A., U. Guin, J. Sarkis, and J. H. Lambert. "Decision model with quantification of buyer-supplier trust in advanced technology enterprises." *Benchmarking: An International Journal* (2021).

Conrow E. Risk analysis considerations for space systems. *In AIAA Space 2007 Conference & Exposition.* 2007. https://doi.org/10.2514/6.2007-6025

Costello T., Forkin, E., Paulen, C. Roy, T., Swierczewski, M., Loose D.C., Williams, J.Y., Slutzky, D., Polmateer T., Manasco, M.C., Hendrickson, D., Lambert, J.H. “Capacity planning and investment into electrification of container ports” *2023 Systems and Information Engineering Design Symposium (SIEDS)*. Accepted paper. Publishing May 2023.

Crater PDC, “2021 comprehensive economic development strategy,” Crater Planning District Commission, Petersburg, VA 23805, Tech. Rep., June 2021.

Crater Planning District Commission (CPDC), “About us,” https://craterpdc.org/about-us/, accessed: September 2021.

Cutler, M., Walsh, T. J., & How, J. P. “Reinforcement learning with multi-fidelity simulators.” In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3888-3895). IEEE. (2014, May).

Cyberseurity and Infrastructure Security Agency (CISA). *Alert (AA22-083A): tactics, techniques, and procedures of indicted state-sponsored russian cyber actors targeting the energy sector;* Cyberseurity and Infrastructure Security Agency (CISA): Washington, DC, USA, 2022

Darwish, A. and Hassanien, A.E. “Cyber physical systems design, methodology, and integration: the current status and future outlook.” *Journal of Ambient Intelligence and Humanized Computing*, 2018. *9*, pp.1541-1556.

Davis, G. B. "Strategies for information requirements determination." *IBM systems journal* 21, no. 1 (1982): 4-30.

DeMarco, T. "Structure analysis and system specification." In *Pioneers and Their Contributions to Software Engineering*, pp. 255-288. Springer, Berlin, Heidelberg, 1979.

Department of Homeland Security (DHS). “Effectiveness of the program to prepare communities for complex coordinated terrorist attacks and the countering violent extremism grant program”. *Fiscal Year 2017 Report to Congress*. 2018.

Donnan, R. C., C.R. Edwards, A. R. Iyer, T. Karamete, P.F. Myers, S. E. Olson, R.S. Prater, J. H. Lambert *et al.* "Enterprise resilience of maritime container ports to pandemic and other emergent conditions." In *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 1-6. IEEE, 2020.

Duvaud, W. and Hainaut, A. "MuZero general: open reimplementation of MuZero". *Github Repository*. Github. 2020. Retrieved from: https://github.com/werner-duvaud/muzero-general

Eddy, T. L., D.C. Loose, T. L. Polmateer, M. C. Manasco, and J. H. Lambert. "Why system requirements must influence the selection of causal models to address congestion in freight logistics." In *2022 IEEE International Systems Conference (SysCon)*, pp. 1-7. IEEE, 2022.

Eddy, T.L., D. C. Loose, M. C. Manasco, D. Hendrickson, T. L. Polmateer, and J. H. Lambert. " Dynamics of user requirements in logistics networks with pandemic and workforce disruptions." In *2022 Wiley Systems Engineering Journal – Special Issue*. Under Review, expected publication 2023.

Fotuhi, F., Huynh, N., Vidal, J. M., & Xie, Y. "Modeling yard crane operators as reinforcement learning agents." *Research in Transportation Economics*, *42*(1), 3-12. (2013).

Fresno County Rural Transit Agency (FCRTA), "Public transit planning," https://www.fresnocog.org/ project/public-transit-planning/, Fresno, CA, November 2021.

Gacek, C. G., D.J. Gimbel, S. J. Longo, B. I. Mendel, G. N. Sampaio, . L. Polmateer, M.C. Manasco, D. C. Hendrickson, T. L. Eddy, and J. H. Lambert. "Managing operational and environmental risks in the strategic plan of a maritime container port." In *2021 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 1-6. IEEE, 2021.

Goldman, H., McQuaid, R., Picciotto, J. "Cyber resilience for mission assurance." *In Proceedings of the 2011 IEEE International Conference on Technologies for Homeland Security* (HST 2011), Waltham, MA, USA, 15–17 November 2011.

Grynspan, R. "Review of maritime transport: navigating stormy waters". Quote in *United Nations Conference on Trade and Development Report*. 2022

Guan, Y., Xiao W-Q, Cheung RK, Li C-L. "A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis". *Oper Res Lett*. 2002 Oct 1;30(5):343–50.

Güven, C., & Eliiyi, D. T. "Trip allocation and stacking policies at a container terminal." *Transportation Research Procedia*, *3*, 565-573. (2014).

Haimes, Y. Y., S. Kaplan, and J. H. Lambert. "Risk filtering, ranking, and management framework using hierarchical holographic modeling." *Risk Analysis* 22, no. 2 (2002): 383-397.

Hamdy, I. H, M. J. St. John, S. W. Jennings, T.R. Magalhaes, J. H. Roberts, T. L. Polmateer, M.C. Manasco, J.Y. Williams, D. C. Hendrickson, T. L. Eddy, D. C. Loose, M. Chowdhury, J H. Lambert. "Quantum computing and machine learning for efficiency of maritime container port operations." *IEEE Systems and Information Engineering Design Symposium (SIEDS)*. May 2022.

Hassan, R., & Gurning, R. O. S. "Analysis of the container dwell time at container terminal by using simulation modelling." *International Journal of Marine Engineering Innovation and Research*, *5*(1). (2020).

Hassler, M. L., D. J. Andrews, B. C. Ezell, T. L. Polmateer, and J. H. Lambert. "Multi-perspective scenario-based preferences in enterprise risk analysis of public safety 5G wireless broadband network." *Reliability Engineering & System Safety* 197 (2020): 106775.

Hatzi, O., Vrakas, D., Nikolaidou, M., Bassiliades, N., Anagnostopoulos, D. and Vlahavas, I., 2011. "An integrated approach to automated semantic web service composition through planning." *IEEE Transactions on Services Computing*, *5*(3), pp.319-332.

He, Y. L., Zhang, X. L., Ao, W., & Huang, J. Z. "Determining the optimal temperature parameter for Softmax function in reinforcement learning." *Applied Soft Computing,* (2018). 70, 80-85.

Helo, P., Paukku, H., and Sairanen, T. “Containership cargo profiles, cargo systems, and stowage capacity: Key performance indicators.” *Maritime Economics & Logistics.* 2018https://doi.org/10.1057/s4127 8-018-0106-z.

Hendrickson, D. and Green, R., “Personal communication to pi’s,” Personal Communication, Norfolk, VA, October 2021.

Hennig, A., T. G. Topcu, and Z. Szajnfarber. “So you think your system is complex?: Why and how existing complexity measures rarely agree.” *Journal of Mechanical Design*, 144(4), 2022.

Hill, R., D. C. Loose, and J.H. Lambert. “Assessment of hybrid and compounding threats in Virginia.” Unpublished Manuscript, expected publication 2023.

Hollenback, K., J. H. Lambert, and I. Linkov. "Indicators and metrics of emerging country-level STEM innovation." *IEEE Engineering Management Review* 48, no. 2 (2020): 47-53.

Hull, E., K. Jackson, and J. Dick. “Requirements engineering in the solution domain.” *Springer London*, 2005.

Hutchins, G. “ISO 31000: 2018 enterprise risk management.” Greg Hutchins; 2018.

IEEE Computer Society. Software Engineering Technical Committee (IEEE CS). “IEEE guide for developing system requirements specifications”. 1998. IEEE.

International Organization for Standardization (ISO). “Robots for industrial environments — automatic end effector exchange systems — vocabulary” *ISO/DIS Standard 1593:2022*. (2022). Retrieved from https://www.iso.org/standard/74676.html

Jennings, M. D. "Gap analysis: concepts, methods, and recent results." *Landscape ecology* 15, no. 1 (2002): 5-20.

Jensen, P. D., and C. Orfila. "Mapping the production-consumption gap of an urban food system: an empirical case study of food security and resilience." *Food security* 13, no. 3 (2021): 551-570.

Jimena, P., D. Aranda, F. Hidalgo, A. E. Smith, E. Karakaya, and R. G.González-Ramírez. "Empty container stacking operations: case study of an empty container depot in Valparaiso Chile." In *2016 Winter Simulation Conference (WSC)*, pp. 3724-3725. IEEE, 2016.

Jin, A. S., L. Hogewood, S. Fries, J. H. Lambert, L. Fiondella, A. Strelzoff, J. Boone, K. Fleckner, and I. Linkov. "Resilience of cyber-physical systems: role of AI, digital twins and edge computing." *IEEE Engineering Management Review* (2022).

Jung, W., & Jazizadeh, F. "Comparative assessment of HVAC control strategies using personal thermal comfort and sensitivity models." *Building and Environment*, *158*, 104-119. (2019).

Kawahara, R., Dotan, D., Sakairi, T., Ono, K.; Nakamura, H., Kirshin, A., Hirose, S., Ishikawa, H. "Verification of embedded system's specification using collaborative simulation of SysML and Simulink models." *In Proceedings of the 2009 International Conference on Model-Based Systems Engineering*, Haifa, Israel, 2–5 March 2009.

Keisler, J. M., Z. A. Collier, B. M. Ayyub, C. S. Dempwolf, J. MacDonald Gibson, A. L. Porter, V. J. Schweizer, J. H. Lambert *et al.* "Modeling and analytics to support emerging international innovation partnerships." *IEEE Engineering Management Review* 48, no. 2 (2020): 54-64.

Kemme, N. "State-of-the-art yard crane scheduling and stacking." *Handbook of Terminal Planning* (2020): 383-413.

Khalil, S.M., Bahsi, H., Ochieng'Dola, H., Korõtko, T., McLaughlin, K. and Kotkas, V., "Threat modeling of cyber-physical systems-a case study of a microgrid system." *Computers & Security*, 2023, *124*, p.102950.

Kim, K.H., Ju Woo, Y., and Kim, J.G. "Space reservation and remarshalling operations for outbound containers in marine terminals." *Maritime Economics & Logistics*. 2019. https://doi.org/10.1057/s4127 8-019-00125 -7.

Klasa, K., B. D. Trump, I. Linkov, and J. H. Lambert. "Identifying new partnerships for innovation: governance and policy challenges." *IEEE Engineering Management Review* 48, no. 2 (2020): 26-38.

Kuzmanović, M. "Bounded rationality in decision making models." In *Conference proceedings-XLVI International Symposium on Operational Research SYMOPIS.* 2019, November. (Vol. 46, p. 5). FON.

Laarni, J., Koskinen, H., & Väätänen, "A concept of operations as a boundary object for knowledge sharing in the design of robotic swarms." *International Journal of Robotics and Control Systems*. (2022). *2*(4), 692-708.

Lambert, J.H., Polmateer T.L., Loose, D.C., Eddy T.L,. Moghadasi N., Johnson D., Manasco M.C. "Industry supply chain disruption analysis for Crater planning district commission". *Commonwealth Center for Advanced Logistics Systems*. 2022.

Larsen, R. and Pacino, D. "-". *Maritime Economics & Logistics*. 2020. https ://doi.org/10.1057/s4127 8-020-00172-5

Lee, Y., and N. Y. Hsu. "An optimization model for the container pre-marshalling problem". *Computers & Operations Research*, 2007. 34:3295–3313

Leveson, N. "A new accident model for engineering safer systems." *Saf. Sci.* **2004**, *42*, 237–270.

Linkov, I., B. D. Trump, J. M. Keisler, J. Cegan, C. M. Foran, Z. A. Collier, J. H. Lambert, and M. M. Kuklja. "Signals and metrics identifying partnerships for innovation." *IEEE Engineering Management Review* 48, no. 2 (2020): 39-46

Liu, P., and Li, Y. "An improved failure mode and effect analysis method for multi-criteria group decision-making in green logistics risk assessment." *Reliability Engineering & System Safety* 215 (2021): 107826.

Liu, X.F. "Fuzzy requirements". *IEEE Potentials*, 1998. 17(2), pp.24-26.

Liu, Y., Yan, S., Zhao, Y., Song, C. and Li, F. "Improved dyna-q: a reinforcement learning method focused via heuristic graph for path planning in dynamic environments". *Drones*, 2022. *6*(11), p.365.

Long, C.K., Van Hai, P., Tuan, T.M., Lan, L.T.H., Ngan, T.T. and Chuan, P.M., "." *Engineering Applications of Artificial Intelligence*, *120*, 2023. p.105920.

Loose, D. C., T. L. Eddy, T. L. Polmateer, D. C. Hendrickson, N. Moghadasi, and J. H. Lambert. "Reinforcement learning and automatic control for resilience of maritime container ports" *IEEE Conference on Control, Decision, and Information Technologies*. Under Review. 2023

Loose, D. C., T. L. Eddy, T. L. Polmateer, M. C. Manasco, N. Moghadasi, D. K. Morris, and J. H. Lambert. "Resilience of sociotechnical systems to evolving priorities" In *2023 Wiley Systems Engineering Journal – Special Issue*. Under Review, expected publication 2023.

Loose, D. C., T. L. Eddy, T. L. Polmateer, M. C. Manasco, N. Moghadasi, D. K. Morris, and J. H. Lambert. "Equity and quantification of risk as disruption of system orders of an industrial region". *Wiley Risk Analysis*. Under Review. 2023.

Loose, D. C., T.L. Eddy, T. L. Polmateer, M. C. Manasco, N. Moghadasi, and J. H. Lambert. "Managing pandemic resilience with other cascading disruptions of a socio-technical system." In *2022 IEEE International Systems Conference (SysCon)*, pp. 1-6. IEEE, 2022.

Loose, D. C., T.L. Eddy, T. L. Polmateer, M. C. Manasco, N. Moghadasi, and J. H. Lambert. "A mathematical framework for tracking resilience of sociotechnical systems and other disruptions." Presentation. *Eighth International Engineering Systems Symposium CESUN 2021*. October 12, 2021.

Loose, D.C., Polmateer T.L., Eddy T.L, Moghadasi N., Johnson D., Manasco M.C., Lambert J.H. "Industry supply chain disruption analysis for Crater planning district commission". Presentation to Crater Planning District Commission. 2022.

Majumder, S., E. Aghayi, M. Noferesti, H. Memarzadeh-Tehran, T. Mondal, Z. Pang, and M. J. Deen. "Smart homes for elderly healthcare—recent advances and research challenges." *Sensors* 17, no. 11 (2017): 2496.

Maldonado, S, R. G. González-Ramírez, F. Quijada, and A. Ramírez-Nafarrate. "Analytics meets port logistics: A decision support system for container stacking operations." *Decision Support Systems* 121 (2019): 84-93.

Martin, J.N. "Systems engineering guidebook: A process for developing systems and products." *CRC Press*, 2020.

Marttunen, M. and Mustajoki, J. "Use of analyst-generated stakeholder preference profiles in multi-criteria decision analysis—Experiences from an urban planning case." *Journal of Environmental Assessment Policy and Management*, *20*(03), p.1840002. 2018.

McDermott, T., Clifford, M.M., Sherburne, T., Horowitz, B. and Beling, P.A. “Framework for operational resilience in engineering and system test (forest) part i: methodology–responding to “security as a functional requirement”. 2022. *INSIGHT*, *25*(2), pp.30-37.

Mclean, E. L., and A. Becker. "Decision makers’ barriers to climate and extreme weather adaptation: a study of North Atlantic high-and medium-use seaports." *Sustainability Science* 15, no. 3 (2020): 835-847.

Mineraud, J., O.Mazhelis, X. Su, and S. Tarkoma. "A gap analysis of Internet-of-Things platforms." *Computer Communications* 89 (2016): 5-16.

Mittal, S., and S. A. Cane. "Contextualizing emergent behavior in system of systems engineering using gap analysis." In *Proceedings of the Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2016 (MSCIAAS 2016) and Space Simulation for Planetary Space Exploration (SPACE 2016)*, pp. 1-9. 2016.

Moerland, T. M., Broekens, J., Plaat, A., & Jonker, C. M. “Model-based reinforcement learning: A survey.” *Foundations and Trends in Machine Learning*, *16*(1), 1-118. (2023).

Moghadasi, N., Z. A. Collier, A. Koch, D. L. Slutzky, T. L. Polmateer, M. C. Manasco, and J.H. Lambert. "Trust and security of electric vehicle-to-grid systems and hardware supply chains." *Reliability Engineering & System Safety* (2022): 108565.

National Center for Immunization and Respiratory Disease (NCIRD), “Guidance for covid-19 prevention in k-12 schools,” *Center for Disease Control and Prevention*, Tech. Rep., August 2021.

Oroojlooyjadid, A., Nazari, M., Snyder, L.V. and Takáč, M., “A deep q-network for the beer game: Deep reinforcement learning for inventory optimization.” *Manufacturing & Service Operations Management*, *24*(1), 2022. pp.285-304.

Pascual, J., D. Aranda, F. Hidalgo, A. E. Smith, E. Karakaya, and R. G. González-Ramírez. "Empty container stacking operations: case study of an empty container depot in Valparaiso Chile." In *2016 Winter Simulation Conference (WSC)*, pp. 3724-3725. IEEE, 2016.

Pennetti, C. A., D. Andrews, M. D. Porter, and J.H. Lambert. “Traveler perception of transportation system performance using kernel density estimation to prioritize infrastructure investments." In *International Conference on Transportation and Development 2020*, pp. 48-61. Reston, VA: American Society of Civil Engineers, 2020.

Pennetti, C. A., D. J. Andrews, T. L. Polmateer, and J. H. Lambert. "Geospatially distributed safety and performance benefits for projects of a transportation system." In *2020 IEEE International Systems Conference (SysCon)*, pp. 1-7. IEEE, 2020.

Pennetti, C. A., J. Jun, G. S. Jones, and J. H. Lambert. "Temporal disaggregation of performance measures to manage uncertainty in transportation logistics and scheduling." *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 7, no. 1 (2021): 04020047.

Pennetti, C. A., S.Sreekumar, K. Hollenback, M. D. Fontaine, and J. H. Lambert. "Quantifying operational disruptions as measured by transportation network reliability." *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 6, no. 4 (2020): 04020039.

Pennetti, C.A., M. D. Fontaine, J. Jun, and J. H. Lambert. "Evaluating capacity of transportation operations with highway travel time reliability." *Reliability Engineering & System Safety* 204 (2020): 107126.

Personal correspondence, Daniel Hendrickson, Vice President: Asset Management and Special Projects at The Port of Virginia. February 8, 2023.

Pireva, K., Kefalas, P. and Cowling, A., “A review of automated planning and its application to cloud e-learning.” 2017

Port of Los Angeles (PoLA). “Logistics optimization with quantum: port of Los Angeles.” Webinar. May 18, 2022.

Port of Virginia (PoV). “2022 State of the port: operations fluid, $1.3b of investments coming, preparing for expansion”. The Port of Virginia, Norfolk, VA. April 2022.

Port of Virginia (PoV). “Facilities”. *Virginia Port Authority* website. 2023. Retrieved from https://www.portofvirginia.com/facilities/.

Port of Virginia (PoV). “Operations functional specifications: Norfolk international terminals optimization project”. The Port of Virginia, 2018

Powell, W.B. “Designing lookahead policies for sequential decision problems in transportation and logistics.” *IEEE Open Journal of Intelligent Transportation Systems*, *3*, (2022) pp.313-327.

Quenum, A., H. Thorisson, D. Wu, and J. H. Lambert. "Resilience of business strategy to emergent and future conditions." *Journal of Risk Research* 24, no. 7 (2021): 870-888.

Rabe, M., Gonzalez-Feliu, J., Chicaiza-Vaca, J. and Tordecilla, R.D. “Simulation-optimization approach for multi-period facility location problems with forecasted and random demands in a last-mile logistics application.” *Algorithms*, *14*(2), (2021) p.41.

Rei, R.J., Kubo, M., Pedroso, J.P, “Simulation-based optimization for steel stacking.” In *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences*. Springer, pp. 254–263. (2008)

Sadiq, M., Lin, C.Y., Wang, K.T., Trung, L.M., Duong, K.D. and Ngo, T.Q. "Commodity dynamism in the COVID-19 crisis: Are gold, oil, and stock commodity prices, symmetrical?." *Resources policy*, *79*, p.103033. 2022.

Schmitt, A. J., & Singh, M. "Quantifying supply chain disruption risk using Monte Carlo and discrete-event simulation." In Proceedings of the *2009 winter simulation conference (WSC)* (2009, December). (pp. 1237-1248). IEEE.

Schrittwieser, J., I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez *et al.* "Mastering Atari, go, chess and shogi by planning with a learned model." *Nature* 588, no. 7839 (2020): 604-609.

Semega, J. and Killar, M. "Increase in income inequality driven by real declines in income at the bottom." *United States Census Bureau (US Census)*. 2022.

Sgobbi, F. and Codara, L. "Resilience capability and successful adoption of digital technologies: two case studies." In *Resilience in a Digital Age* (pp. 309-327). Springer, Cham. 2022.

SHIPA Freight. "10 largest ports in North America". *Shipa Knowledge Series.* 2023. Retrieved from https://www.shipafreight.com/knowledge-series/largest-ports-in-north-america/

Sikorra, J., Bohlken, W., Goes, M. "Allocation of container slots based on machine learning." *Hhla. De*, 11-15. (2021).

Simio. "The story of Simio". Simio LLC. Retrieved from https://www.simio.com/about-simio/. 2022

Sindre, G., Opdahl, A.L. "Eliciting security requirements with misuse cases." *Requirements Engineering* 2005, *10*, 34–44.

Statista. "U.S. wealth distribution Q3 2022". *Statistia Research Department*. February 1, 2023.

Stern, J. L., P. T. Grogan, and A. Valencia-Romero. “Strategic robustness in bi-level system-of-systems design.” 2021.

Suhail, S., Malik, S. U. R., Jurdak, R., Hussain, R., Matulevičius, R., & Svetinovic, D. “Towards situational aware cyber-physical systems: A security-enhancing use case of blockchain-based digital twins.” *Computers in Industry*, *141*, 103699. (2022).

The White House. “Fact sheet: Biden-Harris administration announces additional infrastructure funding for ports and waterways.” *White House Press Briefing*. Washington, DC. March 2022.

The White House. “Building resilient supply chains, revitalizing American manufacturing, and fostering broad-based growth.” *Executive Order 14-17*. Washington, DC. June 2021.

The White House. “Remarks by president Biden on inflation and actions taken to lower prices and address supply chain challenges.” *White House Briefing Room Speeches and Remarks*. Los Angeles, California. June 10, 2022.

The World Bank. Sustainable development - building a balanced future. https://www.worldbank.org/en/who-we-are/news/ campaigns/2020/the-sustainable-development-series-building-a-balanced-future, 2022. Accessed: March 2022.

Thomas, P. J., G. Perono, F.Tommasi, G. Pagano, R. Oral, P. Burić, I. Kovačić, M. Toscanesi, M. Trifuoggi, and D. M. Lyons. "Resolving the effects of environmental micro-and nanoplastics exposure in biota: A knowledge gap analysis." *Science of the Total Environment* 780 (2021): 146534.

Thorisson, H., and J. H. Lambert. "On the “influence of scenarios to priorities” in risk and security programs." *Applied Risk Analysis for Guiding Homeland Security Policy* (2021): 1-24.

Thorisson, H., C. A. Pennetti, D. J. Andrews, D.C. Hendrickson, T. L. Polmateer, and James H. Lambert. "Systems modeling and optimization of container ship berthing with various enterprise risks." In *2019 IEEE International Systems Conference (SysCon)*, pp. 1-8. IEEE, 2019.

Thorisson, H., F. Baiardi, M. Salminen, R. Van Erdeghem, R.Sahay, B.Paquin, C. Heath, J. H. Lambert *et al.* "Cyber security challenges to arctic critical infrastructures." *Cybersecurity and Resilience in the Arctic* 58 (2020): 60.

Thorisson, H., M. Alsultan, D.Hendrickson, T. L. Polmateer, and J. H. Lambert. "Addressing schedule disruptions in business processes of advanced logistics systems." *Systems Engineering* 22, no. 1 (2019): 66-79.

Tracy, K.W. *Software: A Technical History*. Morgan & Claypool, 2021.

Turton, W. and Mehrotra, K., “Hackers breached colonial pipeline using com- promised password,” https://www.bloomberg.com/news/articles/2021-06-04/ hackers-breached-colonial-pipeline-using-compromised-password, June 2021, accessed: September 2021.

U.S. Army, “U.S. army garrison fort Lee,” https://home.army.mil/lee/index.php, accessed: September 2021.

U.S. Bureau of Labor Statistics (USBLS), “Employment and wages, annual averages 2020,” U.S. Bureau of Labor Statistics, Washington, DC 20212-0001, Tech. Rep., September 2021.

U.S. Coast Guard (USCG) “Maritime severe weather contingency plan”. *US Coast Guard Sector Hampton Roads*. June 2019

UN News. “Ports, shipping need to go green to resist future global crises: UNCTAD.” *UN News Global perspective human stories*. November 2022.

United Nations Conference on Trade and Development (UNCTAD). “Review of maritime transport: navigating stormy waters”. *United Nations Report.* 2022

United Nations Department of Economic and Social Affairs (UNDESA). “Transforming our world: the 2030 Agenda for Sustainable Development”. *Sustainable Development.* 2022. Retrieved from https://sdgs.un.org/2030agenda

United Nations Environment Programme (UNEP), UNEP Copenhagen Climate Centre (UNEP-CCC). “Emissions gap report 2021 – key messages.” *Emissions gap report 2021*. October 2021.

United Nations System Chief Executives Board for Coordination (United Nations CEB). “UN system framework for action on equality.” *32nd Session of the High-level Committee on Programmes (Geneva).* March 2016. Revised 2022. Retrieved from https://unsceb.org/un-system-framework-action-equality

United States Census Bureau (US Census). “Wealth inequality in the U.S. by household type”. 2022.

van Asperen, E., Borgman, B., & Dekker, R. (2013). Evaluating impact of truck announcements on container stacking efficiency. *Flexible Services and Manufacturing Journal*, *25*, 543-556.

Van Lamsweerde, A. "Requirements engineering in the year 00: A research perspective." In *Proceedings of the 22nd international conference on Software Engineering*, pp. 5-19. 2000.

VanYe, C.M., B. E. Li, A.T. Koch, M.N. Luu, R. O. Adekunle, N. Moghadasi, Z. A. Collier, J. H. Lambert *et al.* "Trust and security of embedded smart devices in advanced logistics systems." In *2021 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 1-6. IEEE, 2021.

Virginia Department of Emergency Management (VDEM), D.C. Loose *et al*. "Commonwealth of Virginia hazard mitigation plan". *Virginia Department of Emergency Management*. 2023.

VDEM. "Commonwealth of Virginia hazard mitigation plan". Virginia Department of Emergency Management. 2018.

Verasetti. "How to write a system specification". February 10, 2021. Retrieved from: https://www.verasseti.com/insights/how-to-write-a-system-specification#:~:text=What%20is%20a%20System%20Requirement,of%20the%20Software%20Development%20Lifecycle.

Virginia Code Commission (VCC), Code of Virginia, Section 15.2-4207, 2021

Virginia Department of Conservation and Recreation (DCR). "The natural communities of Virginia classification of ecological groups and community types". *Natural Heritage.* March 2021.

Virginia Economic Development Partnership (VEDP), "Virginia business ready sites program (vbrsp)," https://www.vedp.org/vbrsp, Richmond, VA, January 2021.

Virginia Employment Commission (VEC), Economic Information Analytics Division, "Local area un- employment statistics (laus) - August 2021," Virginia Employment Commission, Richmond, VA 23230, Tech. Rep., September 2021.

Virginia Port Authority. "Annual comprehensive financial report for fiscal year ended June 30, 2021". The Port of Virginia, 2021

Vlahavas, I. and I. Refanidis, "Planning and scheduling technical report" (EETN, 2013)

Wang, N., Chang, D., Shi, X., Yuan, J., & Gao, Y. “Analysis and design of typical automated container terminals layout considering carbon emissions.” *Sustainability*, *11*(10), 2957. (2019).

Weilkiens, T. “Systems engineering with SysML/UML: modeling, analysis, design”. Elsevier, 2011.

Wheeler, R. L., C. A. Pennetti, T. L. Polmateer, G. Jones, Y. Dhanpal, and J. H. Lambert. "Information management of a lifeline infrastructure for mobility of people and goods." In *2022 IEEE International Systems Conference (SysCon)*, pp. 1-4. IEEE, 2022.

World Bank Group (WBG). “Transport global practice; the container port performance index”. *S&P Global Market Intelligence*. 2022.

Wu, D. D., J. Mitchell, and J. H. Lambert. " Global systemic risk and resilience for novel coronavirus in postpandemic era." *Risk Analysis* 42 (1), 1-4.

Wu, D. D., J. Mitchell, and J. H. Lambert. "Global systemic risk and resilience for novel coronavirus and covid-19." *Risk Analysis* 41, no. 5 (2021): 701-704.

Yan, Y., Chow, A. H., Ho, C. P., Kuo, Y. H., Wu, Q., & Ying, C. “Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities.” *Transportation Research Part E: Logistics and Transportation Review*, *162*, 102712. (2022).

Yang, C.C. and Hsu, W.L., “Evaluating the impact of security management practices on resilience capability in maritime firms—A relational perspective.” *Transportation Research Part A: Policy and Practice*, *110*, pp.220-233. 2018.

Yilmaz, E., O. Sanni, M. T. Kotwicz Herniczek, and B.German. "Deep reinforcement learning approach to air traffic optimization using the muzero algorithm." In *AIAA AVIATION 2021 FORUM*, p. 2377. 2021.

Zachman, J. A. "A framework for information systems architecture." *IBM systems journal* 26, no. 3 (1987): 276-292.

Zhan, A., P. Zhao, L. Pinto, P. Abbeel, and M. Laskin. "A framework for efficient robotic manipulation." *arXiv preprint arXiv:2012.07975* (2020).

Zhang, S., & Sutton, R. S. “A deeper look at experience replay.” (2017). arXiv preprint arXiv:1712.01275.

Appendices

A.1 Overview

There are two sections in the Appendices. Section A.2 provides the complete list of gaps identified during the hazard and gap analysis of Chapter 4. Section A.3 provides the full code used to develop the system simulation of Chapter 5.

A.2 Gaps

Drought

Pennsylvania Drought Resiliency Program	
Hazards Addressed:	Drought
Description:	A drought resiliency program focused on maintaining national and federal partnerships for drought management. The program centralizes several resources for residents and businesses, including links to the National Drought Resilience Partnership, EPA reports on water use and droughts, and funds for drought resilient water infrastructure projects. Drought status is tracked via a number of metrics including groundwater levels and precipitation. Resilience is measured using ability to persist during drought and recover after an event. Mitigations include conservation and improving access.
Origin:	Pennsylvania
Hazard(s) Addressed:	Drought
Lead Agency:	Department of Environmental Protection
Funding:	
Status:	Ongoing
Comments:	Data sources for drought status, precipitation, stream flow, ground water, burn bans, and soil moisture are provided.
Further Details: https://www.dep.pa.gov/Business/Water/PlanningConservation/Drought/Pages/Drought-Resiliency.aspx#:~:text=Drought%20resiliency%20focuses%20on%20preparing,help%20communities%20become%20more%20resilient.	

North Carolina Water Supply Planning	
Description:	A supply planning and demand management resource to coordinate local decisions at a state level. Local governments must prepare water supply plans and submit water usage data through an online portal. Current drought status for each region is provided along with recommended actions for each severity. Providing locality specific recommendations is expected to improve response compared to statewide recommendations. The threshold for increasing drought severity is a quarter of the land area in the region.
Origin:	North Carolina
Hazard(s) Addressed:	Drought
Lead Agency:	Department of Environmental Protection
Funding:	
Status:	Ongoing
Comments:	537 municipalities, counties, and other regional entities publish plans for water supply and shortage response.
Further Details: https://deq.nc.gov/about/divisions/water-resources/water-planning/water-supply-planning/water-use-reporting	

California Department of Water Resources Drought Funding	
Description:	The California Department of Water Resources provides funding for projects that combat the impacts of drought conditions. Eligible applicants include counties named in drought emergencies or in areas needing immediate action, public agencies and utilities, special districts, colleges and universities, Native American tribes and others. Of the authorized funds, 40% is authorized for small communities. Factors considered include human health and safety, fish and wildlife, and loss of supply. Funds for the 2022 cycle must be allocated by 2024 and projects completed by 2026.
Origin:	California
Hazard(s) Addressed:	Drought
Lead Agency:	California Department of Water Resources
Funding:	
Status:	Ongoing
Comments:	The department of water resources is also developed a methodology to assess local vulnerability and risk scoring along with recommendations for contingency planning in at risk areas.
Further Details: https://water.ca.gov/Water-Basics/Drought/Drought-Funding	

Earthquake

Alaska Seismic Hazards Safety Commission	
Description:	Recommends goals and priorities for seismic risk mitigation. The commission partners with federal organizations to identify schools that are at-risk of damage in the event of an earthquake. The commission outlines plans to perform seismic retrofits on at-risk buildings. Alaska also receives FEMA funding for earthquake resilience projects, including retrofitting the Port of Alaska, the construction of earthquake-resistant water transmission lines, grid and gas updates, and gas shut off valves. Remaining funds are allocated to schools and other critical facilities as part of the retrofit program
Origin:	Alaska
Hazard(s) Addressed:	Earthquake
Lead Agency:	Office of the Governor: Alaska Seismic Hazards Safety Commission
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://seismic.alaska.gov/ https://dggs.alaska.gov/webpubs/dggs/ic/text/ic088.pdf	

Earthquake Warning California	
Description:	California participates in ShakeAlert, an earthquake early warning system for the West coast.
Origin:	California
Hazard(s) Addressed:	Earthquake
Lead Agency:	USGS
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.shakealert.org/	

Brace & Bolt Program	
Description:	The California Office of Emergency Services provides up to $3,000 to cover retrofitting costs for homeowners through the Brace and Bolt program. The California Residential Mitigation Program also provides grants and financial assistance to low income and vulnerable populations to retrofit and harden homes as mitigation for earthquakes.
Origin:	California
Hazard(s) Addressed:	Earthquake
Lead Agency:	California Residential Mitigation Program (CRMP)
Funding:	
Status:	Ongoing
Comments:	The California Earthquake Authority, a not-for-profit partnered with the Cal OES, provides earthquake insurance policies for homeowners and renters.
Further Details: https://www.earthquakebracebolt.com/	

Debris Removal Services	
Description:	Local governments are eligible for assistance with debris removal through the California Consolidated Debris Removal Program. Teams will inspect property and remove hazardous materials that pose a threat to human health and the environment.
Origin:	California
Hazard(s) Addressed:	Earthquake
Lead Agency:	California Office of Emergency Services
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.placer.ca.gov/DocumentCenter/View/55030/Debris-Removal-Program-Enrollment-and-Process-FAQ-2021?bidId=	

Erosion

Maryland Stormwater, Dam Safety, and Flood Management Program (SDSFM)	
Description:	"The Program Review division of SDSFM manages the stormwater and sediment and erosion control programs… The Plan Review Division reviews construction plans on State and federal projects for consistency with Stormwater Management regulations (SWM) and Erosion and Sediment Control (ESC) regulations, then issues approval." The SDSFM issues permits and directs local governments to reduce pollution and erosion from runoff due to construction activities. The SDSFM delegates oversight to some local governments and manages other regions directly.
Origin:	Maryland
Hazard(s) Addressed:	Erosion
Lead Agency:	Maryland Department of the Environment;
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://mde.maryland.gov/programs/Water/SSDS/Pages/index.aspx#:~:text=The%20SDSFM%20program%20consists%20of,and%20Compliance%2C%20and%20Flood%20Management.&text=The%20Program%20Review%20division%20of,sediment%20and%20erosion%20control%20programs.	

Massachusetts StormSmart Coasts Program	
Description:	The StormSmart Coasts Program "provides information, strategies, and tools to help communities and people working and living on the coast to address the challenges of erosion, flooding, storms, sea level rise, and other climate change impacts." The program provides tools for homeowners and local officials regarding strategies for reducing coastal erosion and storm damage while reducing impacts to shorelines. The program maintains the Barrier Beach Inventory Project, which maintains data on barrier beaches including recent changes, developments, damage, and other information critical for barrier beach management. The program includes the Coastal Resilience Grant program, which addresses challenges caused by sea level rise, storms, flooding, and erosion.
Origin:	Massachusetts
Hazard(s) Addressed:	Erosion
Lead Agency:	Massachusetts Office of Energy and Environmental Affairs, through the Office of Coastal Zone Management
Funding:	
Status:	Ongoing
Comments:	Other projects include planning, redesigns and retrofits, and shoreline restoration.
Further Details: https://www.mass.gov/stormsmart-coasts-program#:~:text=The%20Massachusetts%20Office%20of%20Coastal,and%20other%20climate%20change%20impacts.	

California Erosion Control Toolbox	
Description:	California provides an erosion control toolbox through Caltrans (California DOT), which provides “Landscape Architects with a single location that contains the information necessary to design successful, cost-effective and sustainable erosion control treatments”
Origin:	California
Hazard(s) Addressed:	Erosion
Lead Agency:	Caltrans (California DOT)
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://dot.ca.gov/programs/design/lap-erosion-control-design/tool-1-lap-erosion-control-toolbox#:~:text=The%20purpose%20of%20the%20Caltrans,Erosion%20Control%20Treatments https://dbw.parks.ca.gov/?page_id=28766	

Public Beach Restoration and Shoreline Erosion Program	
Description:	The California Division of Boating and Waterways controls the Public Beach Restoration and Shoreline Erosion Program, which issues grants for the repair, redevelopment, and hardening of beaches at risk of major erosion. The Public Beach Restoration and Shoreline Erosion Program also provides workshops for potential applicants. The program will provide experts to survey shorelines regarding suitability for new projects.
Origin:	California
Hazard(s) Addressed:	Erosion
Lead Agency:	California Division of Boating and Waterways
Funding:	
Status:	Ongoing
Comments:	The program and its projects are beholden to standards outlined in the California Harbors and Navigation Code – Department of Boating and Waterways Code (ARTICLE 2.5. Beach Erosion Control [65 - 67.4])
Further Details: https://www.grants.ca.gov/grants/division-of-boating-and-waterways-public-beach-restoration-program/ https://dbw.parks.ca.gov/?page_id=28766	

Extreme Cold

Pennsylvania Warming Centers	
Description:	Pennsylvania Department of Human Services provides information on a network of extreme cold warming centers through partnerships between the state, counties, and non-profit organizations. The service includes an interactive map and searchable database that provides information on warming center locations, hours, eligibility, intake procedures, capacity limits, alerts, and contact information.
Origin:	Pennsylvania
Hazard(s) Addressed:	Extreme Cold
Lead Agency:	Pennsylvania Department of Human Services
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.pa211.org/get-help/housing-shelter/extreme-cold-warming-centers/	

Baltimore Code Blue	
Description:	Baltimore manages the Code Blue Extreme Cold program. When a Code Blue is declared, several responses are triggered to protect individuals experiencing homelessness. City-funded shelters will shelter-in-place to ensure any individual experiencing homelessness and wanting shelter will be accommodated. Private homeless shelters will be encouraged to extend their hours and keep individuals indoors.
Origin:	Maryland
Hazard(s) Addressed:	Extreme Cold
Lead Agency:	Baltimore City Health Department
Funding:	
Status:	Ongoing
Comments:	Homeless Services Outreach Workers provide cold weather education, encourage individuals experiencing homelessness to take shelter, and connect them to services as needed. On nights when Code Blue Extreme Cold has been declared, the Salvation Army FEEDMORE canteen provides hot drinks and other items to individuals experiencing homelessness.
Further Details: https://health.baltimorecity.gov/emergency-preparedness-response/code-blue#:~:text=A%20Code%20Blue%20Extreme%20Cold%20declaration%20triggers%20several%20responses%20aimed,wanting%20shelter%20will%20be%20accommodated.	

Extreme Heat

Virginia Regional Efforts	
Description:	Richmond, Southside Hampton Roads, and Northern Virginia have participated in urban heat island mapping exercises, outlining neighborhoods that are vulnerable to extreme heat conditions. These projects were undertaken by the cities themselves and are included in the U.S. Climate resilience toolkit
Origin:	Virginia
Hazard(s) Addressed:	Extreme Heat
Lead Agency:	Multiple
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://toolkit.climate.gov/case-studies/where-do-we-need-shade-mapping-urban-heat-islands-richmond-virginia https://www.adaptationclearinghouse.org/resources/citizen-science-mapping-urban-heat-islands-in-richmond-virginia.html#:~:text=The%20urban%20heat%20island%20mapping,design%20community%2Dscale%20adaptation%20plans.	

Philadelphia Beat the Heat	
Description:	Many cities in Pennsylvania, including Philadelphia, provide toolkits and research to communities with recommendations on how to take community-level actions to combat the effects of extreme heat. The toolkits help communities in Philadelphia research their part of the city, establish heat teams, interview stakeholders, conduct surveys, organize the community, designate "Beat the Heat" ambassadors, create mobile stations, host workshops, promote tree planting, and build a heat relief network.
Origin:	Pennsylvania
Hazard(s) Addressed:	Extreme Heat
Lead Agency:	City of Philadelphia Office of Sustainability
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.phila.gov/departments/office-of-sustainability/beat-the-heat-toolkit/	

Maryland Extreme Heat Emergency Plan	
Description:	Maryland utilizes an Extreme Heat Emergency Plan, which outlines sets of triggering events, organizations responsible for surveillance, and organizations responsible for actions. The primary organizations are the Maryland Department of Health and local health departments. Summers are divided into six phases: Pre-Summer, Pre-Event, Extreme Heat Event – Heat Advisory, Extreme Heat Event – Excessive Heat Warning, Complex Heat Emergency, and Post-Summer. During each phase, various organizations have different responsibilities. Responsibilities include timings for press releases, reports on heat and water use, responses to events such as power outages or water shortages, and actions to take post-event.
Origin:	Maryland
Hazard(s) Addressed:	Extreme Heat
Lead Agency:	Maryland Department of Health
Funding:	
Status:	Ongoing
Comments:	As extreme heat conditions worsen, state organizations mobilize various efforts to reduce harm to citizens. For example, local health departments will notify state agencies of hotspots in order to distribute resources during a heat advisory. During a Complex Heat Emergency (a heat event compounded by other factors such as power outages), the Maryland Department of Health will take an advisory role on outage plans and coordinate local emergency services.
Further Details: https://health.maryland.gov/preparedness/Documents/MDH%20Extreme%20Heat%20Emergency%20Plan%202022.pdf	

Massachusetts Extreme Heat Resources	
Description:	Many localities within Massachusetts have strategies for managing extreme heat. These may include the distribution of cooling care kits, fans, AC units, and wearable cooling devices. This is orchestrated at the community level with sponsorship from the state and regional planning councils. Strategies have been updated to accommodate Covid safety. Analysis of neighborhood vulnerability to extreme heat was assessed and results are available to support planning.
Origin:	Massachusetts
Hazard(s) Addressed:	Extreme Heat
Lead Agency:	Metropolitan Area Planning Council
Funding:	
Status:	Ongoing
Comments:	Hotels and motels are converted into safe areas for residents without in-home cooling. These organizations are compensated for temporarily housing affected individuals and families.
Further Details: https://www.mapc.org/resource-library/extreme-heat-resources/	

North Carolina Climate and Health Program	
Description:	The US CDC has issued a grant to North Carolina through the Climate-Ready States and Cities Initiative to operate the North Carolina Climate and Health program. The program "aims to serve elementary school students, farmworkers, local public health preparedness and emergency management staff, low-income earners, older adults requiring nutritional support, and young adults attending county parks".
Origin:	North Carolina
Hazard(s) Addressed:	Extreme Heat
Lead Agency:	North Carolina Department of Health and Human Services
Funding:	
Status:	Ongoing
Comments:	The program has implemented heat-related illness syndromic surveillance and heat health alert systems in several counties across the state.
Further Details: https://www.cdc.gov/climateandhealth/climate_ready.htm	

New York City Heat Island Mapping	
Description:	New York City has participated in heat island mapping exercises. Factors included air pollution, human health, and nighttime cooling. There is greater risk to those at risk for severe Covid symptoms. After the heat island mapping exercise, NYC launched the NYC CoolRoofs program, an effort to train workers and install energy-saving reflective rooftops. The program also funds street tree planting in vulnerable neighborhoods.
Origin:	New York
Hazard(s) Addressed:	Extreme Heat
Lead Agency:	New York City Council
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://council.nyc.gov/data/heat/	

California Extreme Heat Worker Protections	
Description:	California utilizes legislation to protect workers from extreme heat and heat related illness while on the job. The law includes provisions for workers that mandate easy access to water, access to shade, new procedures for high heat days, emergency procedures, acclimatization, and training. Partial exemptions are provided for a number of industries including agriculture, construction, landscaping, and oil and gas extraction. Employers create an effective heat illness prevention plan. High-heat procedures are required above 95 degrees.
Origin:	California
Hazard(s) Addressed:	Extreme Heat
Lead Agency:	California Division of Occupational Safety and Health
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.dir.ca.gov/title8/3395.html	

Flooding

Pennsylvania Flood Planning	
Description:	The Pennsylvania Department of Environmental Protection: Bureau of Waterways Engineering and Wetlands provides financial and technical assistance to various municipal sponsors to reduce the impact of floods or prevent floods entirely. The program includes funding for investigation of areas at risk for flooding and evaluating long-term solutions to flooding. This includes assessments of the magnitude and frequency of flooding, performing hydraulic analysis, evaluating flood control alternatives, estimating costs, assessing environmental impacts, performing a cost/benefit analysis, defining sponsors, and beginning to preparing designs. Protections may include concrete channels, concrete floodwalls, compacted earthen levees, channel improvements, or other alternatives Pennsylvania law encourages every community to participate in NFIP. Communities that choose not to participate in NFIP are ineligible for hazard mitigation non-disaster funds or other disaster funding streams.
Origin:	Pennsylvania
Hazard(s) Addressed:	Flooding
Lead Agency:	Pennsylvania Department of Environmental Protection
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.dep.pa.gov/Business/Water/Waterways/Flood-Protection/Pages/default.aspx	

Florida Flood Protection Level of Service Program	
Description:	Parts of Florida utilize the Flood Protection Level of Service Program. This is an in-depth, regimented program dedicated to prioritizing flood mitigation projects in South Florida. The program utilizes Adaptive Resilience Planning to determine which mitigation actions are appropriate for flood-prone areas considering uncertain future conditions.
Origin:	Florida
Hazard(s) Addressed:	Flooding
Lead Agency:	South Florida Water Management District
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.sfwmd.gov/our-work/flood-protection-level-service#:~:text=The%20FPLOS%20program%20ensures%20a,additional%20%242%20million%20a%20year.	

Florida QuickGuide for Floodplain Management	
Description:	Florida has a QuickGuide available to communities for their use in explaining floodplain management concepts at the permit counter.
Origin:	Florida
Hazard(s) Addressed:	Flooding
Lead Agency:	Florida Division of Emergency Management
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.floridadisaster.org/contentassets/5a671dfdfadf45ab9a2c61635e2a4fed/quick-guide-for-floodplain-management.pdf	

Florida State Assistance Information Hotline	
Description:	The Florida State Assistance Information Line is a toll-free hotline available for residents in the event of a flood, hurricane, or other disaster. Contacting the line will provide information on: "How to prepare before/during/after a hurricane, road closures and alternate routes, available/open shelters in host or impacted counties, shelters designed for special needs patients, hotels and motels that accept pets, boaters instructions for moving watercraft to safer ground, and re-entry information once it is safe to return to the affected area."
Origin:	Florida
Hazard(s) Addressed:	Flooding
Lead Agency:	Florida Division of Emergency Management
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.floridadisaster.org/planprepare/information-line/	

Florida Building Code	
Description:	The Florida Building Code is more strict than NFIP requirements and applicable statewide, regardless of the participation status of a community in the NFIP.
Origin:	Florida
Hazard(s) Addressed:	Flooding
Lead Agency:	Florida Department of Business and Professional Regulation
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://floridabuilding.org/c/default.aspx	

Floodplain Administration Documents	
Description:	Many customized assistance documents for local floodplain administrators (guidance, ordinance amendment language and sample forms) available online from FDEM.
Origin:	Florida
Hazard(s) Addressed:	Flooding
Lead Agency:	Florida Division of Emergency Management
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.floridadisaster.org/dem/mitigation/floodplain/community-resources/	

Hurricanes

Maryland Block Grant Program	
Description:	The Maryland Department of Housing and Community Development provides funds for vulnerable communities in the event of a hurricane through the Community Development Block Grant Disaster Recovery Assistance program. Grants were issued by Congress in response to Hurricane Sandy. Maryland used all funds for recovery activities in the worst hit counties.
Origin:	Maryland
Hazard(s) Addressed:	Hurricanes
Lead Agency:	Maryland Department of Housing and Community Development
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://dhcd.maryland.gov/Communities/Pages/cdbg/CDBGSandyDisasterRecovery.aspx	

North Carolina Hurricane Preparedness	
Description:	The North Carolina Department of Public Safety partners with the National Weather Service to host an annual Hurricane Preparedness Week, a series of meetings and associated resources outlining how to prepare for hurricanes. One focus area of the awareness week is on people with disabilities, how these individuals should prepare for hurricanes, and how to accommodate people with disabilities in the event of evacuations.
Origin:	North Carolina
Hazard(s) Addressed:	Hurricanes
Lead Agency:	North Carolina Department of Public Safety
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.weather.gov/ilm/hurricaneprepnc	

NC State Centric Hazard Mitigation Pilot Program	
Description:	North Carolina also utilizes a State Centric Hazard Mitigation Pilot Program the first of its kind for FEMA HMGP, that allows the state to manage and pay for contract work to complete all grants awarded, and assists the counties by removing the financial and management burden of completing all the work awarded under each grant. What's more, the local government does not lose the management costs paid to the local government under the grant agreement. It only speeds the process for homeowners in need by centralizing the project management.
Origin:	North Carolina, Flooding, Others
Hazard(s) Addressed:	Hurricanes
Lead Agency:	North Carolina Department of Public Safety
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.ncdps.gov/our-organization/emergency-management/disaster-recovery/hazard-mitigation/state-centric-hazard	

Florida Hurricane Program	
Description:	The Florida Division of Emergency Management created the Hurricane Loss Mitigation Program that funds mitigation projects. Projects include retrofits to residential, commercial, and mobile home properties, increased public education programs, and hurricane research activities.
Origin:	Florida
Hazard(s) Addressed:	Hurricanes
Lead Agency:	Florida Division of Emergency Management
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.floridadisaster.org/dem/mitigation/hurricane-loss-mitigation-program/	

Florida Hurricane Program	
Description:	Florida funds the Florida International University International Hurricane Research Center (IHRC). The IHRC focuses on research that reduces hurricane damage and loss of life through more effective mitigation.
Origin:	Florida
Hazard(s) Addressed:	Hurricanes
Lead Agency:	Florida International University
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.ihrc.fiu.edu/	

Impoundment Failure

Private Dam Financial Assurance Program	
Description:	The Pennsylvania Department of Community and Economic Development administers the Private Dam Financial Assurance Program to ensure private dams meet and maintain safety standards. The loan amount cannot exceed 50% of the eligible project costs or $500,000, whichever is less. Eligible dam owners are anyone who owns, controls, operates, maintains, or manages a regulated private dam and is enrolled in the Private Dam Financial Assurance Program. This assistance largely takes the form of low interest loans for eligible dams.
Origin:	Pennsylvania
Hazard(s) Addressed:	Impoundment Failure
Lead Agency:	Pennsylvania Department of Community and Economic Development
Funding:	
Status:	Ongoing
Comments:	Virginia operates a similar program through the Department of Conservation and recreation, but with more strict eligibility requirements than the Pennsylvania program.
Further Details: https://dced.pa.gov/programs/private-dam-financial-assurance-program-pdfap/ https://www.dep.pa.gov/Business/Water/Waterways/DamSafety/Pages/default.aspx https://www.dcr.virginia.gov/dam-safety-and-floodplains/dsfpm-grants	

Pennsylvania Dam Safety Program	
Description:	Pennsylvania DEP includes several mitigation strategies in the Dam Safety Program, including several regulations (The Dam Safety & Encroachments Act; The Pennsylvania Dam Safety and Waterway Management Code; The Run-of-the-River Dam Act, and Hazards on the Water Fact Sheet), dam inspection guidelines and cadence, and permit management.
Origin:	Pennsylvania
Hazard(s) Addressed:	Impoundment Failure
Lead Agency:	Pennsylvania Department of Environmental Protection
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.dep.pa.gov/Business/Water/Waterways/DamSafety/Pages/default.aspx	

Maryland Dam Breach Analysis	
Description:	The Maryland Department of the Environment provides several Dam Breach Analysis resources, including providing modeling software and other programs to outline dam performance and stress testing for dam safety. These include analysis methods for small ponds and dams, earthen dams, flooding, spillway, riser, and other hydrology approaches.
Origin:	Maryland
Hazard(s) Addressed:	Impoundment Failure
Lead Agency:	Maryland Department of the Environment
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://mde.maryland.gov/programs/water/damsafety/pages/dambreakguidelines.aspx	

Massachusetts Dam Removal	
Description:	The Massachusetts Division of Ecological Restoration provides resources to help dam owners remove old, damaged, or outdated dams. The office manages the requirements for dam safety and maintains documentation of dam design documents. Dam owners must submit reports to the Office of Dam Safety. The Massachusetts Division of Ecological Restoration provides resources to help dam owners remove old, damaged, or outdated dams. The MA DER outlines the circumstances in which an individual would want to remove a dam, such as when maintenance costs are too high, legal liability changes, or the cost of repair is greater than the value of the dam. Further considerations are ecological – the DER can assess if a small dam is impacting local water quality as part of its process for determining which projects are funded. Massachusetts offers several programs that can assist with dam removal, including the "Dam and Seawall Repair or Removal Program, the Massachusetts Environmental Trust (MET) Grant Program, the Municipal Vulnerability Preparedness (MVP) Program, and DER's Priority Projects Program." The Priority Projects program is the only grant program administered by the MA DER, but the DER will work with applicants to secure other funding resources as necessary
Origin:	Massachusetts
Hazard(s) Addressed:	Impoundment Failure
Lead Agency:	Massachusetts Office of Dam Safety
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.mass.gov/river-restoration-dam-removal#:~:text=The%20Division%20of%20Ecological%20Restoration,Become%20a%20DER%20Priority%20Project%E2%80%9D.	

Texas Dam Safety	
Description:	The Texas Commission on Environmental Quality manages a Dam Safety Program, specifically Dam Safety Workshops for Owners and Operators. These workshops assist owners and operators with understanding dam safety laws and regulations and enforcement, emergency action plans and maintenance issues for all areas on a dam, recommendations for correction, and results of the probable maximum precipitation study.
Origin:	Texas
Hazard(s) Addressed:	Impoundment Failure
Lead Agency:	Texas Commission on Environmental Quality
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.tceq.texas.gov/compliance/investigation/damsafetyprog.html#:~:text=The%20Dam%20Safety%20Program%20monitors,help%20them%20maintain%20safe%20facilities.	

Karst (Sinkholes)

Pennsylvania Sinkhole Data	
Description:	The Pennsylvania Department of Conservation and Natural Resources provides resources for individuals who may have seen or been affected by a sinkhole. These include educational resources on: geological and human activities contributing to sinkholes, safety, repair, and prevention.
Origin:	Pennsylvania
Hazard(s) Addressed:	Karst (Sinkholes)
Lead Agency:	Pennsylvania Department of Conservation and Natural Resources
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.gis.dcnr.state.pa.us/pageode/ https://www.dcnr.pa.gov/Geology/GeologicHazards/Sinkholes/Pages/default.aspx	

Maryland Geological Survey	
Description:	The Maryland Geological Survey produces a step-by-step guide for individuals outlining what to do if one suspects they have encountered a sinkhole. The guide is available online, distributed through the Maryland Geological Survey newsletter, and in a series of videos posted to the MGS website. The article provides links for reporting sinkholes and contact information for the appropriate state and local agencies.
Origin:	Maryland
Hazard(s) Addressed:	Karst (Sinkholes)
Lead Agency:	Maryland Geological Survey
Funding:	
Status:	Ongoing
Comments:	
Further Details: http://www.mgs.md.gov/geology/geohazards/sinkhole_resources.html	

Landslides

Introduction to Landslides in North Carolina	
Description:	The North Carolina Department of Environmental Quality publishes worksheets and press releases on how to identify and respond to landslide conditions. NCDEQ hosts this information on a webpage that includes links to historical data, information on rock slope stability, and the increased risk of landslides during hurricanes or other severe weather events.
Origin:	North Carolina
Hazard(s) Addressed:	Landslides
Lead Agency:	North Carolina Department of Environmental Quality
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://deq.nc.gov/about/divisions/energy-mineral-and-land-resources/north-carolina-geological-survey/geologic-hazards/landslides	

California Landslide Mapping	
Description:	The California Department of Conservation performs routine landslide mapping activities through the Seismic Hazards Program. New buildings, mines, and other construction activities are required to submit geotechnical reports on the land to the State Geologist. This data is added to the state geotechnical database and hazard maps.
Origin:	California
Hazard(s) Addressed:	Landslides
Lead Agency:	California Department of Conservation
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.conservation.ca.gov/cgs/sh/program#:~:text=The%20Seismic%20Hazards%20Program%20delineates,fault%20rupture%2C%20and%20tsunami%20inundation.	

National Landslide Hazard Mitigation Strategy	
Description:	The United States Geological Survey maintains the "National Landslide Hazards Mitigation Strategy." Recommendations include: • Research - Developing a predictive understanding of landslide processes and triggering mechanisms • Hazard mapping and assessments - Delineating susceptible areas and different types of landslide hazards at a scale useful for planning and decision making • Real-time monitoring - Monitoring active landslides that pose substantial risk • Loss assessment - Compiling and evaluating information on the economic impacts of landslide hazards • Data Collection - Information collection, interpretation, and dissemination • Guidelines and training - Developing guidelines and training for scientists, engineers, and decisionmakers • Public awareness and education - Developing information and education for the user community • Implementation of loss reduction measures - Encouraging mitigation action • Emergency preparedness, response, and recovery - Building resilient communities
Origin:	Federal
Hazard(s) Addressed:	Landslides
Lead Agency:	United State Geological Survey
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.conservation.ca.gov/cgs/sh/program#:~:text=The%20Seismic%20Hazards%20Program%20delineates,faul%20rupture%2C%20and%20tsunami%20inundation.	

Land Subsidence

Pennsylvania Mine Subsidence Insurance	
Description:	Pennsylvania offers Mine Subsidence Insurance. Residential Coverage of $150,000 costs $41.25 a year. Depending on subsidence risk levels, coverage of up to $1,000,000 is available. The Pennsylvania Department of Environmental Protection administers the MSI program, providing an online portal to connect consumers with insurance providers. Private insurers apply to the PA DEP to become registered sellers of mine subsidence insurance. The MSI program publishes sales kits of the individual insurance providers and provides tips & tools to the insurance providers to maximize the return of sales efforts. The program provides information to homeowners, indicating which properties are located on top of abandoned mines
Origin:	Pennsylvania
Hazard(s) Addressed:	Land Subsidence
Lead Agency:	The Pennsylvania Department of Environmental Protection
Funding:	Agency Funds, Cost Sharing
Status:	Ongoing
Comments:	
Further Details: https://www.dep.pa.gov/Citizens/MSI/Pages/default.aspx	

Maryland Land Subsidence Monitoring Network	
Description:	The Maryland Geological Survey maintains the Land Subsidence Monitoring Network, a service that monitors land subsidence in at-risk areas of Maryland, especially near the Chesapeake Bay. The focus of the program is on isolating vertical land motion attributed to human activities (such as groundwater withdrawal). Data is gathered and analyzed annually.
Origin:	Maryland
Hazard(s) Addressed:	Land Subsidence
Lead Agency:	The Maryland Geological Survey, USGS
Funding:	
Status:	Ongoing
Comments:	
Further Details: http://www.mgs.md.gov/groundwater/current/land_subsidence.html	

California Aqueduct Subsidence Program	
Description:	The California Department of Water Resources manages the California Aqueduct Subsidence Program. The program studies areas at risk of subsidence due to aqueduct levels as part of the State Water Project. This projected yielded reports on areas at risk of damage due to aqueduct subsidence.
Origin:	California
Hazard(s) Addressed:	Land Subsidence
Lead Agency:	The California Department of Water Resources
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://water.ca.gov/Programs/Engineering-And-Construction/Subsidence	

Non-Tornadic Wind

Nebraska Severe Weather Preparedness Guide	
Description:	The Nebraska Emergency Management Agency releases a Spring and Summer Severe Weather Preparedness Guide. While this largely focuses on thunderstorms and tornados, it also presents tips for other high-wind events such as how to identify and take action before a thunderstorm.
Origin:	Nebraska
Hazard(s) Addressed:	Non-Tornadic Wind
Lead Agency:	The Nebraska Emergency Management Agency
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://nema.nebraska.gov/operations/spring-and-summer-severe-weather-preparedness	

Farming in Challenging Times Roundtable	
Description:	The Farming in Challenging Times roundtable recommends maintaining up-to-date insurance policies, paying particular focus to the age and structural integrity of properties. Further considerations include proximity to trees, fences, and electrical wiring.
Origin:	Private Sector
Hazard(s) Addressed:	Non-Tornadic Wind
Lead Agency:	Nationwide Insurance
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.agweb.com/news/business/farmland/derecho-response-wind-and-storm-mitigation-your-farm	

NOAA Guidelines	
Description:	NOAA recommends that individuals set aside emergency supplies such as food, water, batteries, and flashlights when high-wind conditions are expected. Further, power outages and infrastructure damage may make it difficult to reach gasoline for transportation and to power generators.
Origin:	Federal
Hazard(s) Addressed:	Non-Tornadic Wind
Lead Agency:	National Oceanic and Atmospheric Administration
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.spc.noaa.gov/misc/AbtDerechos/derechofaq.htm	

Center for Disaster Philanthropy	
Description:	The Center for Disaster Philanthropy provides information on how to recover from high-wind and derecho conditions. Low-cost short and long-term housing is needed to support those with no familial support in affected areas. Mental health services are necessary to support the long-term resilience of affected regions.
Origin:	Private Organization
Hazard(s) Addressed:	Non-Tornadic Wind
Lead Agency:	The Center for Disaster Philanthropy
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://disasterphilanthropy.org/disasters/midwest-derecho/	

Iowa State University Lessons Learned	
Description:	Iowa State University provides a set of lessons-learned from derecho tree breaks and has outlined recommendations for preventing and reducing damage due to treefalls. When planting trees on a new site, select species that are native to the area. Engage in proper tree-pruning practices to increase tree strength and health. Utilize proper planting techniques, including site prep, correct depth of planting, appropriate planting times, and post-planting maintenance activities. Constantly assess the health and quality of trees, inspecting for damage, disease, or other abnormalities.
Origin:	Iowa
Hazard(s) Addressed:	Non-Tornadic Wind
Lead Agency:	Iowa State University
Funding:	
Status:	Ongoing
Comments:	Virginia may consider utilizing these recommendations as instructions for residents and other organizations.
Further Details: https://www.spc.noaa.gov/misc/AbtDerechos/derechofaq.htm	

Pandemic

Maryland Communicable Disease Program	
Description:	The Maryland Department of Health Infectious Disease Bureau utilizes the Communicable Disease Program, which provides free immunizations, screenings, and treatments to eligible populations. These services are managed at the state level but administered by local health departments. The program also conducts disease surveillance and provides educational resources.
Origin:	Maryland
Hazard(s) Addressed:	Pandemics
Lead Agency:	Maryland Department of Health, Infectious Disease Bureau
Funding:	
Status:	Ongoing
Comments:	The Virginia Department of Health provides several similar services, including immunization for tuberculosis and some STIs. Some major gaps include HIV treatment and care, Adult Viral Hepatitis prevention, and a center for Zoonotic and Vector-borne Diseases.
Further Details: https://health.maryland.gov/phpa/pages/infectious-disease.aspx	

Massachusetts Disease Control & Prevention Resources	
Description:	The Massachusetts Health & Social Services Disease Control and Prevention program maintains a set of resources available to residents, distributed online and through pamphlets given to healthcare providers. Resources include information and fact sheets on infectious diseases, data on flu seasons, Asthma risks, tick-borne diseases, risk factor surveillance, and information on cancer & cancer screenings.
Origin:	Massachusetts
Hazard(s) Addressed:	Pandemics
Lead Agency:	Massachusetts Department of Health & Social Services
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.mass.gov/topics/disease-control-prevention	

North Carolina Safety Net Dental Clinics	
Description:	The North Carolina Department of Health and Human services maintains a list of Safety Net Dental Clinics for low-income individuals. Eligibility varies across clinics. Covered services include fluoride mouth rinse and dental sealant projects; dental assessments, screenings, and referrals; education services; and consultation services. Further, the program includes a focus on perinatal oral health to improve the overall standard of care during pregnancy. Additionally, the N.C. Oral Health Section helped local agencies to expand and maintain their Safety Net Dental Clinics.
Origin:	North Carolina
Hazard(s) Addressed:	Pandemics
Lead Agency:	North Carolina Department of Health and Human Services, Division of Public Health
Funding:	
Status:	Ongoing
Comments:	While not a direct pandemic mitigation, increasing access to care and reducing disparities in health is a way to mitigate the impacts of a pandemic scenario. Virginia has Safety Net clinics, including dental clinics, but lacks the targeted publicity campaign of North Carolina.
Further Details: https://www.dph.ncdhhs.gov/oralhealth/services/safety-net.htm https://www.vhcf.org/who-and-how-we-help/medical/health-safety-net-providers/	

Tornado

Pennsylvania StormReady Participation	
Description:	The National Weather Service administers the StormReady service, which outlines a set of activities that a county must perform to attain StormReady Status, helping communities establish plans of action to prevent damage and recover from severe weather, including tornadoes. Pennsylvania has pushed for all counties in the state to reach the "StormReady" status, and is one of six states to have all counties participating. PEMA actively encourages cities and counties to participate in the program.
Origin:	Pennsylvania
Hazard(s) Addressed:	Tornado, Hurricanes, Flooding, Non-Tornadic Wind
Lead Agency:	National Weather Service, Pennsylvania Emergency Management Agency
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.weather.gov/StormReady https://www.ready.pa.gov/pages/stormready.aspx	

Oklahoma SoonerSafe Safe Room Rebate	
Description:	The Oklahoma Office of Emergency Management operates the SoonerSafe Safe Room Rebate program that provides reimbursement for homeowners that install tornado shelters. The safe rooms may be installed in new or existing homes, in interior rooms or under the first floor of the home, or a detached above-ground safe room within 100 feet of the home. The program is funded through HMGP funds, with a maximum rebate of $2,000 and not exceeding 75% of the actual cost of the safe room. Oklahoma law allows for 100 sq. ft. of new safe room to be exempt from property taxation. Only residential single-family homes are eligible; mobile home owners are eligible for single safe room only.
Origin:	Pennsylvania
Hazard(s) Addressed:	Tornado
Lead Agency:	Oklahoma Department of Emergency Management and Homeland Security
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://oklahoma.gov/oem/programs-and-services/soonersafe-safe-room-rebate-program.html	

Wildfires

Community Wildfire Prevention Grants Program	
Description:	The Wildfire Prevention Grants Program seeks to reduce the risk factors associated with wildfires. The grants can be used to clear debris and brush, reduce the presence of other hazardous fuels, develop community plans for wildfire mitigation, and provide educational materials. Organizations or communities may apply for grants for hazardous fuels reduction, wildfire prevention planning, wildfire education, and forest health revitalization projects.
Origin:	California
Hazard(s) Addressed:	Wildfires
Lead Agency:	The Department of Forestry and Fire Protection
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.fire.ca.gov/grants/wildfire-prevention-grants/	

Colorado Strategic Wildfire Action Program (COSWAP)	
Description:	COSWAP funds workforce development for combating wildfires, the creation of State Wildland Inmate Fire Teams (SWIFT), investment into landscape resilience, projects, and establishing state, local, and commercial partnerships to fund future mitigation projects. The COSWAP program funds workforce development grants for training teams to prevent and combat wildfires. COSWAP also assesses the state for Strategic Focus Areas – regions of Colorado that are at particularly high risk of wildfire damage. Colorado Correctional Industries (CCi) contributes wildland fire teams. The SWIFT crews are housed at various correctional facilities in Colorado. Currently, the base locations are at the Four Mile Correctional Center in Canon City, the Rifle Correctional Center in Rifle and the Buena Vista Correctional Center in Buena Vista, Colorado. CCi makes SWIFT crews available to Colorado State Forest Service (CSFS) and other agencies to assist in fighting fires within Colorado by dispatch through normal dispatch centers. CSFS has routinely provided a crew liaison when crews have been dispatched to wildland fires. The crews are self-sufficient and come with supervisors, basic tools and equipment, and transportation. To ensure that the crews function well, the personnel train together and are maintained as crews throughout the year. They are available year-round for assistance with non-fire, woods-related programs and projects.
Origin:	Colorado
Hazard(s) Addressed:	Wildfires
Lead Agency:	Colorado Department of Natural Resources, Colorado Correctional Industries
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://dnr.colorado.gov/divisions/forestry/co-strategic-wildfire-action-program	

Winter Weather

New York Winter Preparedness Guide	
Description:	The New York Winter Preparedness Guide is an online resource that provides information on energy pricing, consumer protections, tips for managing heating costs, tips for energy conservation, and data about winter safety such as how to properly use heaters. The tool also provides links to other services such as efficient energy programs, home energy assessments, federal and local bill assistance programs, community workshops, and legal assistance.
Origin:	New York
Hazard(s) Addressed:	Winter Weather, Extreme Cold
Lead Agency:	New York Department of Public Service
Funding:	
Status:	Ongoing
Comments:	The guide focuses on the legal rights of residents to regarding utility shutoffs, limits to fuel price changes, and payment programs.
Further Details: https://www3.dps.ny.gov/W/AskPSC.nsf/All/2A2468643DFEC059852581CB005C16A8?OpenDocument	

Wisconsin Electronic Disease Surveillance System	
Description:	The Wisconsin Department of Health Services administers the Wisconsin Electronic Disease Surveillance System (WEDSS), a web-based syndromic surveillance system that collects and processes data from several clinical systems. This system assists public health officials to better assess the impacts of, for example, cold snaps and infectious disease. This improves resource deployment during disaster events. The program also funds interactive courses for public health staff, clinical laboratories, clinics, and other disease reporters.
Origin:	Wisconsin
Hazard(s) Addressed:	Pandemics, Winter Weather, Extreme Cold
Lead Agency:	Wisconsin Department of Health Services
Funding:	
Status:	Ongoing
Comments:	Health care providers are legally compelled to report any patient they treat who is suspected of having a communicable disease.
Further Details: https://www.dhs.wisconsin.gov/wiphin/wedss.htm	

California Wildfire Mitigation Program (CWMP)	
Description:	The California Office of Emergency Services administers the California Wildfire Mitigation Program. Homes at risk of wildfire damage are eligible for grants to be hardened and retrofitted to resist wildfires. This includes building with flame resistant materials, redeveloping land to resist fire, and development of defensible space. Homeowners apply for the grant online. Socially vulnerable populations such as residents over 65, in poverty, living with disabilities, with limited English, or without vehicles are prioritized. Homes in high-risk areas are also given precedence over lower-risk homes.
Origin:	California
Hazard(s) Addressed:	Wildfires
Lead Agency:	California Office of Emergency Services, California Department of Forestry and Fire Protection
Funding:	
Status:	Ongoing
Comments:	
Further Details: https://www.caloes.ca.gov/cal-oes-divisions/recovery/disaster-mitigation-technical-support/california-wildfire-mitigation-program	

A.3 Simulation Code

There are several simulation models of the stacking environment. The model presented here aligns with the primary simulation model presented in Chapter 6.

```
import gym
import numpy as np
import random
import time
import pathlib
import datetime
import torch
from scipy.stats import truncnorm
from .abstract_game import AbstractGame

class Container:


   def __init__(self, ID, peak = 600, mu=100, sigma=25, maxruntime=1200,
baseStack=False):

      # The ID of the container
      self.id = ID

      # mean dwell time
      self.mu = mu
      self.sigma = sigma

      # The fixed expected dwell time, values based on existing research
      self.departInterval = mu

      # Containers arrive uniformly, or according to another distribution
      # Containers that begin in the stacks have variable arrivals
      if baseStack == True:
         self.arrive = np.random.randint(-120, maxruntime/8)
         #self.arrive = np.random.randint(2,10)
      else:
         self.arrive = np.random.randint(2, .8*maxruntime)
         #self.arrive = round(np.random.triangular(0, peak, maxruntime))

      # Containers have an expexted dwell time
      self.expectedDeparture = self.arrive + self.departInterval

      # The actual departure time of a container
      noise = round(np.random.normal(mu, sigma))

      # Cannot depart before it arrives
      while noise < 2:
         noise = round(np.random.normal(mu, sigma))
      self.departure = self.arrive + noise

      # Cannot depart before simulation begins
      while self.departure <= 0:
         #self.departure = np.random.randint(2, 20)
```

```
        self.departure = np.random.randint(-10, 20) + round(np.random.normal(mu,
sigma))

      # The position of the container
      self.position = [0,0]

      # Flag for arrive
      self.place = False

      # Flag for depart
      self.remove = False

   def setPosition(self, column, height):
      self.position = [column, height]

   def setArrive(self, arrival):
      self.arrive = arrival

   def setDepart(self, depart):
      self.departure = depart

   def getDeparture(self, depart):
      return self.departure

class ContainerStack:

   def __init__(self, length=12, height=5, incoming=25, seed=111793, maxruntime=1200):
      np.random.seed(seed)

      self.length = length
      self.height = height
      self.stack = []
      self.maxruntime = maxruntime

      ID = 0

      #Generate the initial container stack at ~50% full
      for i in range(length):
         self.stack.append([])
         k=0
         for j in range(height):
            if i == 0 or i == (length-1):
               pass
            else:
               if np.random.random() >= 0.5 and not (i == length-3 and j == height-1):
                  container = Container(ID, baseStack=True)
                  container.setPosition(i, k)
                  self.stack[i].append(container)
                  ID = ID+1
                  k = k + 1
               else:
                  pass

      #Generate the incoming containers
      for i in range(incoming):
         container = Container(ID)
         container.setPosition(0, i)
         self.stack[0].append(container)
         ID = ID+1

      # Used to reset the stack
      self.stackTuple = tuple(tuple(i) for i in self.stack)
```

```
        #Ascribe departure times to each container, add logic to departures
        np.random.seed()
        for i in self.stack:
            for j in i:
                noise = round(np.random.normal(j.mu, j.sigma))
                while noise < 2:
                    noise = round(np.random.normal(mu, sigma))

                if (j.arrive + noise) <= 0:
                    while (j.departure) <= 0:
                        j.setDepart(np.random.randint(-10, 20) + round(np.random.normal(j.mu,
j.sigma)))
                else:
                    j.setDepart(j.arrive + noise)

    def resetStack(self):
        self.stack = list(list(i) for i in self.stackTuple)
        np.random.seed()
        for i in self.stack:
            for j in i:
                noise = round(np.random.normal(j.mu, j.sigma))
                while noise < 2:
                    noise = round(np.random.normal(j.mu, j.sigma))
                if (j.arrive + noise) <= 0:
                    while (j.departure) <= 0:
                        j.setDepart(np.random.randint(-10, 20) + round(np.random.normal(j.mu,
j.sigma)))
                else:
                    j.setDepart(j.arrive + noise)

    def moveContainer(self, position, destination):

        if position[0] == 0:
            # pop the postion of the container with coordinates in position
            # currentposition = place in list of position in argument
            for i in self.stack[0]:
                if i.position == position:
                    currentposition = self.stack[0].index(i)
                    break

            tempContainer = self.stack[0].pop(currentposition)
            tempContainer.setPosition(destination, len(self.stack[destination]))
            self.stack[destination].append(tempContainer)

        elif len(self.stack[position[0]]) != 0:
            tempContainer = self.stack[position[0]].pop()
            tempContainer.setPosition(destination, len(self.stack[destination]))
            self.stack[destination].append(tempContainer)

        else:
            pass

    def setArrived(self, simTime):
        for i in self.stack[0]:
            if i.arrive <= simTime:
                i.place = True

    def getArrived(self):
        arrived = []
        for cont in self.stack[0]:
            if cont.place == True:
                arrived.append(cont)
        return arrived
```

```
   def setDepart(self, simTime):
      for i in self.stack:
         for j in i:
            if j.departure <= simTime:
               j.remove = True

   def getDepart(self):
      departed = []
      for i in self.stack[:-1]:
         for j in reversed(i):
            if j.remove == True:
               departed.append(j)
      return departed

   def showStack(self, factor):
      view = [[getattr(j, factor) for j in i] for i in self.stack]
      print(*view, sep = "\n")
      print("\n")

   def Moves(self):
      action_set = {}
      num = 0

      for i in range(self.length):
         for j in range(self.length):
            if j == 0 or (i == 0 and j == self.length) or (i == self.length-1) or i ==
j:
               pass
            else:
               action_set[num] = [i,j]
               num += 1

      action_set[num] = "pass"

      return action_set

   def validStacks(self):
      # Returns which of the stacks have a spot open
      tempStack = self.stack[1:-1]
      val = []
      stacknum = 1
      for i in tempStack:
         if len(i) < self.height:
            val.append(stacknum)
         stacknum = stacknum + 1
      return val

class Game(AbstractGame):

   def __init__(self, seed=None):
      self.env = Stacks()

   def step(self, action):

      observation, reward, done = self.env.step(action)
      return observation, reward*10, done

   def legal_actions(self):

      return self.env.legal_actions()

   def reset(self):
```

```
      return self.env.reset()

   def render(self):

      self.env.render()
      input("Press enter to take a step ")

class Stacks:

   def __init__(self, length=12):

      #System Time
      self.t = 0

      #Observation Space
      self.stack = ContainerStack(length)

      #action space
      self.action_space = self.stack.Moves()

      # Useful variable
      self.length = self.stack.length

      #coutner for moves
      self.moves = 0

      #Keep a list of containers that need to move, do it in order
      self.needmove = []

   def step(self, action):


      #move if contianer is arriving
      if len(self.stack.getArrived()) != 0:

         #get position of moving container
         #need to pop from this list
         cont = self.stack.getArrived().pop(0)
         self.stack.moveContainer(cont.position, self.action_space[action][1])
         reward = -1
         self.moves = self.moves + 1

      #if anything but no move, step normally
      elif action != list(self.action_space.keys())[-1]:
         self.stack.moveContainer(self.action_space[action],
self.action_space[action][1])
         reward = -1
         self.moves = self.moves + 1

      #if last, do nothing
      else:
         reward = 0


      self.t = self.t + 1

      if self.t >= self.stack.maxruntime:
         done = True

      elif len(self.legal_actions()) == 0:
         return self.get_observation(100), -500000, True
```

```
      else:
         done = False

      return self.get_observation(100), reward, done



   def legal_actions(self):
      self.stack.setArrived(self.t)
      self.stack.setDepart(self.t)

      #set of keys for the moves dictionary
      valid = []

      if any(self.stack.getArrived()) == True and self.stack.getArrived()[0] not in
self.needmove:
         self.needmove.append(self.stack.getArrived()[0])
      elif any(self.stack.getDepart()) == True and self.stack.getDepart()[0] not in
self.needmove:
         templist = self.stack.getDepart()
         templist.sort(key=lambda x: x.departure)
         self.needmove.append(templist[0])
         self.needmove.sort(key=lambda x: x.departure)
      else:
         pass

      if len(self.needmove) > 0:

         self.needmove.sort(key=lambda x: x.departure)

         if self.needmove[0].position[0] == 0:
            for i in range(self.stack.length-1):
               if len(self.stack.stack[i+1]) != self.stack.height:
                  valid.append(i)
            if self.needmove[0].remove == False:
               valid.pop()
            self.needmove.pop(0)

         elif self.needmove[0].departure <= self.t and self.needmove[0].position[1] ==
len(self.stack.stack[self.needmove[0].position[0]])-1:
            start = self.needmove[0].position[0]
            exit = self.length-1

            valid.append(list(self.action_space.values()).index([start,exit]))
            self.needmove.pop(0)

         else:

            origin = []
            validDestination = self.stack.validStacks()
            c = self.needmove[0]

            if c.position[0] not in origin and c.position[0] != self.length-1:
               origin.append(c.position[0])

               #second - legal destinations - checks if the selected cont is ready to
leave
               #remove stacknum to desination unless i is top of the stack (last
element)
            for k in origin:
               for j in validDestination:
                  if k != self.length-1 and j != k:
                     valid.append(list(self.action_space.values()).index([k,j]))
```

```
            self.needmove.pop(0)


            valid = list(set(valid))


        else:
            # everything not at 0 or with 0 containers can move or no action
            for i in range(len(self.action_space)-1):
                if self.action_space[i][0] != 0 and self.action_space[i][1] !=
self.stack.length-1:
                    if len(self.stack.stack[self.action_space[i][0]]) != 0 and
len(self.stack.stack[self.action_space[i][1]]) != self.stack.height:
                        valid.append(i)
            valid.append(i+1)

        return valid

    def render(self):
        print("Simualtion time: ", self.t)
        print("Total Container Moves: ", self.moves)
        self.stack.showStack('id')

    def reset(self):
        self.stack.resetStack()
        self.t = 0
        self.moves = 0
        return self.get_observation(100)

    def get_observation(self, maxheight):
        temp = [[float(1) for j in i] for i in self.stack.stack]
        square = [np.pad(column, (0, maxheight - len(column)), mode='constant',
constant_values=float(0)) for column in temp]
        return np.array([square])
```