ABSTRACT

PIPER, BRIAN ELLIOTT BELL. Optimization Methods for Improving the Resilience of Civil Infrastructure Systems Subject to Natural Hazards. (Under the direction of Ranji Ranjithan.)

Modeling of civil infrastructure systems subject to hazards and disruptions, and subsequent optimization of their performance under such scenarios, is crucial to protect critical lifeline services. This research seeks to develop methods for discovering cost-effective, incremental, stage-wise investment strategies to improve long-term resilience and performance of complex, interdependent systems against future, unknown hazards. An adaptable modeling framework for the civil infrastructure system is developed to represent system components, quantify the services they provide, and the interdependencies among components. Resilience metrics are created that quantify various aspects of system-wide resilience based on the status of components, allowing consideration of system-wide performance after a natural hazard. An illustration of the metrics conducted on realistic data from a coastal community subject to hurricanes shows that the approach can give an assessment of performance in various scenarios. A mathematical programming approach is used to optimally select component investments and improvements according to the serviceability resilience metric.

The introduced metrics and modeling framework are then extended to a stage-wise approach, where investment decisions are made incrementally, as hazard events occur. A stochastic integer programming model for improving resilience is presented for this incremental, stage-wise approach to making decisions. The separation of decisions into multiple stages and consideration of an increased number of hazard scenarios necessitates the implementation of a customized Benders decomposition algorithm for certain resilience metrics. The results indicate that specialized algorithms or meta-heuristics are necessary to find good solutions.

The computational challenges of particular metrics and the desire to simultaneously optimize multiple measures of resilience lead to the development of an evolutionary multiobjective optimization algorithm for diverse solutions, EMODS, to find a population of potential solutions that achieve
two goals. The first goal is to discover Pareto optimal candidates to capture the trade-offs among different measures of resilience. The second goal is to locate alternative solutions that are near-Pareto optimal (i.e., within a given relaxation of Pareto optimality) that are diverse and distinct in the decision space to better handle unmodeled problem characteristics. The notion of solution diversity is defined for multiobjective problems using the concept of substitutability. Various diversity metrics are defined that are necessary for quantifying the different aspects of solution diversity in a multiobjective context. EMODS is compared against other diversity-enhancing EMO algorithms and the results demonstrate that the algorithm is competitive with, and frequently improves on, the other algorithms in terms of the quality of nondominated frontiers and diversity metrics, defined here and in other papers.

EMODS is adapted to apply to problems with discrete decision variables for the civil infrastructure system resilience problem. Experimental runs are conducted for EMODS to optimize multiple resilience metrics of a civil infrastructure system based on the realistic data from the coastal community. The final solution sets obtained by EMODS for two- and four-objective problems are compared with those obtained by the NSGA-II algorithm and a random search. EMODS performs competitively in terms of quality of its nondominated frontiers and outperforms the other methods in regards to the chosen diversity metrics. Additionally, the robustness of the solutions sets obtained by the various methods are analyzed by evaluating final solution sets on a set of alternative hazard scenarios, simulating the real-world situation of prediction errors and unmodeled objectives. The final solution sets obtained by EMODS are shown to exhibit high diversity among substitutable solutions that are near-nondominated and nondominated.
Optimization Methods for Improving the Resilience of Civil Infrastructure Systems Subject to Natural Hazards

by
Brian Elliott Bell Piper

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

Operations Research
Raleigh, North Carolina
2014

APPROVED BY:

Yayha Fathi
E. Downey Brill

Negash Medhin
Ranji Ranjithan
Chair of Advisory Committee
DEDICATION

For my wife, who always believed that I would make it to the end of this, and for my parents, who always supported me.
BIOGRAPHY

Brian E.B. Piper was born in Iowa and grew up in Mankato, Minnesota. He attended Oberlin College, in Oberlin, Ohio and graduated with a double major in theater and mathematics. He planned to be an actor until a fortuitous encounter with an optimization course led him to graduate school in operations research at North Carolina State University in Raleigh. He completed his Master of Science in Operations Research in 2009 under the direction of Dr. Ranji Ranjithan.
# TABLE OF CONTENTS

List of Tables ......................................................................................... vi

List of Figures ......................................................................................... vii

Chapter 1 Introduction ........................................................................... 1

Chapter 2 Resilience Metrics and Decision Models for Civil Infrastructure Systems ........................................................................ 6
  2.1 Background ...................................................................................... 7
    2.1.1 Quantifying Resilience ................................................................. 7
    2.1.2 Interdependent Infrastructure Modeling ....................................... 9
    2.1.3 Decision Support Models ............................................................. 10
  2.2 Civil Infrastructure System Modeling Framework ................................. 12
    2.2.1 System Representation ............................................................... 13
    2.2.2 Hazard Scenarios .................................................................. 14
    2.2.3 Component States .................................................................. 15
    2.2.4 Resilience Metrics .................................................................. 15
  2.3 Numeric Experiments for Metrics ...................................................... 19
  2.4 Optimization Models ....................................................................... 28
    2.4.1 Decision Variables .................................................................. 28
    2.4.2 Dependency Constraints ............................................................ 31
    2.4.3 Objective Function .................................................................. 33
    2.4.4 Serviceability Optimization Model ........................................... 35
  2.5 Numeric Experiments for Optimization Models .................................... 36
  2.6 Observations and Future Work ......................................................... 39

Chapter 3 Stage-wise Improvements of Civil Infrastructure System Resilience ........................................................................ 42
  3.1 Introduction ...................................................................................... 42
  3.2 Background ...................................................................................... 44
    3.2.1 Interdependent CIS Modeling ................................................... 44
    3.2.2 Stochastic Programming Algorithms ......................................... 47
  3.3 CIS Modeling Framework and Multi-Event Extension ......................... 48
    3.3.1 Resilience Metrics .................................................................. 51
    3.3.2 Decision Model Variables and Dependency Constraints ............... 54
    3.3.3 Optimization Model Instance .................................................... 56
  3.4 Solution Approach ........................................................................... 61
    3.4.1 Special Cuts for Benders Decomposition ...................................... 64
  3.5 Numeric Experiments ....................................................................... 69
    3.5.1 Results .................................................................................... 73
  3.6 Observations ................................................................................... 77

Chapter 4 A Multiobjective Evolutionary Algorithm for Enhancing Decision Space Diversity ......................................................... 79
  4.1 Introduction ...................................................................................... 79
4.2 Background .................................................. 81
4.2.1 Related Work ........................................ 82
4.2.2 Measuring Decision Space Diversity ................. 85
4.3 EMODS ..................................................... 89
4.4 Hypervolume Fitness and Selection ......................... 90
4.4.1 Hypervolume–based Fitness ............................ 90
4.4.2 Selection Operator ..................................... 93
4.5 Archive ..................................................... 94
4.6 Numerical Experiments .................................... 97
4.6.1 Metrics .................................................. 99
4.6.2 Final Solution Set Selection ......................... 99
4.6.3 Results ................................................. 100
4.7 Observations ............................................... 108

Chapter 5  Application of EMODS for Improving Civil Infrastructure System Resilience ... 110
5.1 Introduction ................................................ 110
5.2 Background ............................................. 112
5.2.1 CIS Resilience ........................................ 112
5.2.2 Alternative Solutions for Incompletely Modeled Multiobjective Problems ............ 113
5.3 Civil Infrastructure System Modeling Framework 114
5.3.1 Hazard Scenarios .................................... 114
5.3.2 System Representation ............................... 115
5.3.3 Component Status Calculation ..................... 117
5.3.4 CIS Resilience Metrics ............................... 117
5.3.5 Improvement Decisions .............................. 119
5.4 EMODS Description .................................... 121
5.4.1 Solution Encoding and Variation Operators ............ 123
5.4.2 Decision Diversity Metric .......................... 124
5.4.3 Selection Operator .................................. 127
5.4.4 Archiving Procedure ............................... 129
5.5 Numerical Experiments .................................. 131
5.5.1 Parameters and Setup ................................ 134
5.5.2 Experimental Group 1 Results: Two–Objective Problem ....................................... 135
5.5.3 Experimental Group 2 Results: Four–Objective Problem .................................... 140
5.5.4 Evaluation of Solution Robustness .................. 146
5.5.5 Comparison of Selected Solutions ................. 151
5.6 Observations .............................................. 152

Chapter 6  Observations ........................................ 156

References ..................................................... 160
# LIST OF TABLES

| Table 2.1 | Notation | 16 |
| Table 2.2 | Serviceability by Scenario. See text for abbreviation descriptions | 24 |
| Table 2.3 | Notation for the Optimization Models | 29 |
| Table 2.4 | Optimization Model Results | 38 |
| Table 2.5 | Resilience metrics after service optimizations | 39 |
| Table 3.1 | Notation for Multi–Event Resilience Metrics | 49 |
| Table 3.2 | Notation for Multi–Stage Optimization Model | 54 |
| Table 3.3 | Numeric Experiment Settings | 72 |
| Table 3.4 | Metrics Results for Problem Instance A | 73 |
| Table 3.5 | Value of Stochastic Solution (VSS) and Expected Value of Perfect Information (EVPI) for Problem Instance A | 74 |
| Table 3.6 | Model Performance for Problem Instance A | 75 |
| Table 3.7 | Metrics Results for Problem Instance B | 75 |
| Table 3.8 | Value of the Stochastic Solution (VSS) and Expected Value of Perfect Information (EVPI) for Problem Instance B | 75 |
| Table 3.9 | Model Performance for Problem Instance B | 77 |
| Table 4.1 | Test Problem Characteristics | 98 |
| Table 4.2 | Parameters for Metric Calculations | 99 |
| Table 4.3 | Algorithm Settings | 102 |
| Table 4.4 | Hypervolume Results | 103 |
| Table 4.5 | Solow–Polasky Results | 104 |
| Table 4.6 | Omni-Test Clusters Found | 104 |
| Table 4.7 | Shir Diversity Results | 106 |
| Table 4.8 | Average Minimum Results | 106 |
| Table 4.9 | Minimum Distance Results | 107 |
| Table 4.10 | Average Distance from Nearest Objective Results | 108 |
| Table 5.1 | CIS Modeling Framework Notation for Multi–Event Resilience Metrics | 116 |
| Table 5.2 | Decision Distances Between Component Improvement Options | 126 |
| Table 5.3 | Common Parameters and Values for Coastal City Problem | 134 |
| Table 5.4 | Common Operator Parameters for EMODS and NSGA–II | 135 |
| Table 5.5 | Hypervolume Metric Results for Experimental Group 1 | 137 |
| Table 5.6 | Diversity Metric Results for Experimental Group 1 | 139 |
| Table 5.7 | Hypervolume Metric Results for Experimental Group 2 | 142 |
| Table 5.8 | Diversity Metric Results for Experimental Group 2 | 146 |
| Table 5.9 | Correlations between Metrics for Experimental Group 2 | 147 |
| Table 5.10 | Change in Size of Nondominated Fronts under Alternative Hazard Scenarios | 148 |
| Table 5.11 | Objective Function Values of Selected Solutions | 151 |
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Modeling Framework for the Civil Infrastructure System</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Recovery Time Calculation</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Coastal City Road Network and Levee Protection</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Locations of Failed Components in Different Hazard Scenarios</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Recovery operations for a particular scenario. The highlighted color of a link indicates the time period the link was restored. Total restoration time was 124 periods.</td>
<td>25</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Level Plot of Correlation Matrix</td>
<td>26</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>Scatter Plot of Comparing by Scenario Resilience Metric Performances and Number of Failed Components</td>
<td>27</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>Improvement Decisions for Coastal City</td>
<td>41</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Coastal City Road Network and Levee Protection</td>
<td>57</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Constraint Matrix for Demand Satisfaction Optimization Model</td>
<td>62</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Pseudocode for EMODS</td>
<td>90</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Area of shaded region is induced front hypervolume of of solution $a$</td>
<td>91</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Pseudocode for Induced Front Fitness Calculation</td>
<td>92</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Pseudocode of Archive Update Procedure</td>
<td>95</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Hyperboxes used to calculate overlap in decision space</td>
<td>97</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Pseudocode for Final Near–Pareto Set Selection</td>
<td>101</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Sample of solution clusters found by EMODS for Omni–Test problem</td>
<td>105</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Final Nearly–Nondominated Set ($EMODS_{NN}$) for Two–On–One Problem</td>
<td>107</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Modeling Framework for the Civil Infrastructure System</td>
<td>114</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Coastal City Road Network and Levee Protection</td>
<td>115</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Outline of EMODS Algorithm</td>
<td>122</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Crossover Operator for CIS Encoding</td>
<td>125</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Induced front hypervolume of solution $a$</td>
<td>128</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Greedy Construction Heuristic</td>
<td>131</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>Best Interchange Heuristic</td>
<td>132</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Comparison of Performance Metrics by Method for Experimental Group 1</td>
<td>136</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Final Archives and Populations for Experimental Group 1</td>
<td>138</td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>Comparison of Performance Metrics by Method for Experimental Group 2</td>
<td>141</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>Average Hypervolume Convergence of each Method for Experimental Group 2</td>
<td>143</td>
</tr>
<tr>
<td>Figure 5.12</td>
<td>Average Computation Time per Iteration of Each Method</td>
<td>144</td>
</tr>
<tr>
<td>Figure 5.13</td>
<td>Average Computation Time per Iteration of Each Method and Fitness Evaluation Time</td>
<td>145</td>
</tr>
<tr>
<td>Figure 5.14</td>
<td>Change in Size of Nondominated Fronts under Alternative Hazard Scenarios</td>
<td>149</td>
</tr>
<tr>
<td>Figure 5.15</td>
<td>ADNO Diversity Metrics For Solutions from Final Archives</td>
<td>150</td>
</tr>
<tr>
<td>Figure 5.16</td>
<td>Stage 1 Improvement Plans from EMODS$_{ND}$</td>
<td>153</td>
</tr>
<tr>
<td>Figure 5.17</td>
<td>Stage 1 Improvement Plans from EMODS$_{5%}$</td>
<td>154</td>
</tr>
</tbody>
</table>
Modeling of complex systems subject to hazards and disruptions, and subsequent optimization of their performance under such scenarios, is a crucial area for civil infrastructure systems analysis. System performance represents an aggregation of individual component performances, any one of which could be a function of the performance of other components. Systems consisting of components whose performance can be affected by other components are called interdependent systems. Interdependencies in a system greatly increase analysis complexity. This is especially apparent in the context of improving performance of a civil infrastructure system (CIS). A CIS consists of multiple connected, critical subsystems, such as electrical generation and distribution, water treatment and distribution, and transportation, each of which provides lifeline services necessary for a modern society. Unfortunately, the ubiquity of the CIS subjects it to hazards, both natural and anthropogenic, which reduce or eliminate its ability (that is present in well-functioning systems) to support important lifeline services. Resilient performance of the CIS is therefore an utmost priority, where a system is considered resilient if after a hazard it continues to provide services or is quickly able to recover to do so.
Optimizing the design and operation of a CIS to be resilient requires the development of performance functions, or metrics, capable of quantifying system-wide performance. The desired performance goals frequently have no obvious or commonly accepted approach for quantification. For instance, with CIS resilience, quantifying the various aspects of resilience may require different approaches. Furthermore, there is no clear, intuitive method to rank potential improvement projects for a system according to competing, and possible contradictory, metrics to determine which plan may be most beneficial for maintaining sufficient system performance after a hazard. Balancing competing performance metrics in analysis or when making decisions is a difficult task that may be approached through the use of multiobjective optimization methods.

The task of improving performance of interdependent systems is complicated by certain factors. One concern is interdependencies among components, when the services from one component are essential for another component to fully provide lifeline services. With interdependent systems, analyses that separate component level performance from component dependencies may be incomplete enough to preclude meaningful conclusions. For instance, a failure in one component may cause a cascade of failures through other components via a pathway formed from dependencies, resulting in seemingly minor and local failures becoming massive and widespread.

Interdependencies are not the only complicating factor in modeling and improving complex system performance. Future hazards and events are uncertain, both in terms of consequences and likelihood. It is possible to make use of “what-if”, single-scenario analysis, especially in the case of intentional disruptions (such as terrorism), where the magnitude of necessary worst-case preparation is limited by adversary resources and intelligence. Planning protection against natural hazards cannot take advantage of limited adversary resources or assume any intent on the part of hazards. In the context of a CIS subject to natural hazards, scope may vary considerably, from a series of tornadoes that damage but a few power lines in a remote area to the damage seen in New Orleans after Hurricane Katrina. The likelihood of natural hazards is similarly varied as mere days or many years may separate events.

Another issue is balancing resources between post-hazard recovery actions and pre-hazard mit-
igation and preparation actions. Generally, even effective and well-planned recoveries depend on some minimum levels of service being available. There is an inherent challenge in convincing stakeholders to invest in improvements that provide little immediate benefit to protect against a hazard that may not occur. In addition, pre-hazard decisions for the CIS designed to enhance component resistance to hazards may require time-consuming projects. Funding for these projects varies yearly, and during that time period, constraints, goals, and technologies may change. A partially completed task may not meet the new goals, satisfy changed constraints, or effectively employ new technologies. Robust decisions, capable of adapting to evolving environments, must be pursued to optimize complex and interdependent systems.

It should be noted that the environment for these problems is more appropriately called evolving than dynamic. While environmental changes occur over multiple stages, it can be assumed that re-running the optimization model is a possibility after the changes from each stage have manifested. If previous decisions that have been fully implemented define an improvement plan that cannot be adjusted, however, then the system may lose potential resilience gains. Moreover, the existence of multiple definitions and methods of quantifying performance means that a system may measure well for a particular metric, but poorly for others. Multiobjective optimization methods may provide a range of Pareto-optimal solutions representing trade-offs among the objective functions. One issue with these methods is that generally little regard is given to decision space diversity that could allow decision makers to explore alternative plans that better address concerns with evolving and incompletely modeled environments.

This dissertation research seeks to develop methods for discovering robust, incremental, stage-wise strategies to improve the long-term performance of complex, interdependent systems against unknown future hazards. A modeling framework for interdependent and complex systems is developed for the CIS. Components of the system are represented as nodes in a graph and connected with different types of links. Links fall into two categories. The first category of links represents possible pathways of usage for the services provided by the CIS. Links in the second category regulate when the performance of a component affects and is affected by other components. This modeling
abstraction is flexible and can encompass a wide range of modeling necessities.

For CIS performance characterization, metrics are developed and defined for quantifying system-wide performance to capture various aspects of resilience through the aggregation of component performances. These metrics are applied to realistic data to gain insight of system functionality when stressed from multiple disruptions. Performance in hazard scenarios can be aggregated to estimate system performance in future, unknown hazards. The expressions of these metrics also serve as objective functions and constraints for the optimization models used for decision analysis in CIS planning and design.

Two structures for decisions are investigated: a single-stage decision making setting, where all improvements are chosen before hazards occur without regard to the conditions when the decisions are implemented; and a multi-stage process, where chosen upgrades must be prioritized and implemented sequentially. The latter structure uses an incremental approach to making decisions, seeking to gradually change the performance of the system through multiple stages of disbursements and projects. A gradual approach like this is more applicable for the CIS when projects must be prioritized and implemented over a longer horizon in an evolving environment.

Finally, this dissertation research presents a multiobjective evolutionary algorithm (EA) designed to find a population of potential solutions achieving two goals. The first goal of the algorithm is to discover Pareto optimal candidates. These solutions will capture the trade-offs among different performance measures. The second algorithmic goal is to locate solutions that are at least near-Pareto optimal (i.e., within a given relaxation of Pareto optimality) that are diverse and distinct in the decision space. The latter goal will provide alternative solutions that may improve robustness to changing problem characteristics or handling of unmodeled problem characteristics.

The research in this dissertation will be organized into chapters such that each represents a self-contained journal article. In Chapter 2 a CIS modeling framework is developed and presented along with metrics and optimization models for improving resilience. Numeric results demonstrate the validity of the approach when applied to realistic data. In Chapter 3, the single-event optimization and metrics of the previous chapter are extended to multi-event metrics and incremental decisions.
An L-shaped decomposition algorithm is presented to aid efficient computation of solutions. The EMODS (Evolutionary Multiobjective Optimization Algorithm for Diverse Solutions) is developed and presented in Chapter 4, as an EA designed to find diverse and alternative decisions in multiobjective problems. The application of EMODS to a series of test problems shows the algorithm is competitive with, and often superior to, other diversity enhancing evolutionary algorithms. In Chapter 5, EMODS is applied to the realistic CIS constructed in Chapters 2 and 3, with analysis and conclusions about its performance. The ability of a diversity seeking algorithm to enhance the robustness of obtained solutions is evaluated. The dissertation closes with overall observations and directions for future research in Chapter 6.
The ability of the civil infrastructure system (CIS) to support lifeline services after a hazard is critical. The CIS is important to “public safety, health, and welfare” (ASCE ASC (2009)), but designing improvements projects for these concerns can be difficult due to the complex and interconnected nature of infrastructure systems. Frequently, enabling the CIS to support these concerns is considered akin to pursuing resilience, which is defined as the ability of the CIS to continue to provide lifeline services or quickly recovery to provide lifeline services after a hazard. Enhancing resilience can be accomplished by reducing vulnerability, but the vulnerability caused by the connections between CIS parts is increased by the uncertainty surrounding the magnitude, likelihood, and consequences of future hazard events, as well as the current conditions of the system. There is a great need for modeling frameworks capable of 1) quantifying the resilience of the CIS, 2) accurately capturing the effects of interdependencies among various components, and 3) providing decision support to choose engineering improvements that enhance CIS resilience.

Many studies have focused on improving the resilience of the CIS to support the lifelines necessary for general welfare. Yet few have used a system–wide perspective for measuring and increasing
resilience of the whole infrastructure system. Efforts that have been proposed are often limited to
a single infrastructure system, such as a road network or electrical grid, and neglect the effects that
interdependencies have on system performance. Models capable of handling the interdependencies
among separate infrastructure systems often lack optimization capabilities, and CIS optimization
models usually search for optimal post–hazard restoration decisions.

The modeling framework presented here provides a more comprehensive and data–connected
method for quantifying and improving the resilience of the CIS, while explicitly including the effects
from interdependencies and aggregating performance in multiple hazard scenarios. After a review of
other studies, metrics are presented for quantifying CIS resilience and applied to numeric examples
with data taken derived from realistic CIS data. Then optimization models are described with em-
phasis on describing the decisions necessary for the framework and a simple but adaptable method
for explicit interdependencies. Example instances of the optimization models are then applied to the
same CIS data, to demonstrate the strength of the framework.

2.1 Background

Descriptions of past studies are divided into three areas. First, a summary of efforts that have de-
defined resilience and/or described a method for the measurement thereof is presented. Next, inter-
dependent infrastructure modeling efforts are reviewed, especially the mechanisms for representing
interdependency and the capability of the model for optimizing infrastructure resilience. The re-
view then concludes with an analysis of decision support models and computational systems used
to mitigate the risk imposed on the CIS due to hazards.

2.1.1 Quantifying Resilience

The notion of resilience of a system can be found in many areas in scientific literature. Bruneau
et al. (2003) provide a framework to represent resilience through three characteristics: reduced failure
probabilities, reduced consequences, and reduced time to recovery. The resilience of an infrastruc-
ture is measured by calculating the total amount of service lost over the time that the infrastructure remains not fully serviceable. Nearly all studies concerning resilience implicitly include one of the characteristics listed.

For specific infrastructure systems there is a broad range of resilience metrics that have been defined. For general transportation networks, resilience is commonly defined as the percentage of Origin–Destination (OD) pairs for which the demand between them can be met, or the percentage of the total OD demand that can be satisfied (Miller-Hooks et al., 2012). Murray-Tuite (2006) provides ten dimensions for the resilience of a transportation system and measures different aspects of transportation system performance, such as safety and recovery. In the more specific context of road networks, Berdica (2002) provides definitions of vulnerability and serviceability (the latter is adapted in this paper as a quantifiable metric in Section 2.2.4). Scott et al. (2006) describe a network robustness index, which calculates resilience for each link in the road network through what–if analysis of usage pattern changes and travel time increases if each link were to be out of service.

Some research efforts have considered economic resilience, such as Rose (2009), where a framework is developed to consider direct and indirect loses from natural hazards and terrorist attacks. This framework has been used to analyze and quantify the losses from the September 11th terrorist attacks (Rose, 2009) and the effectiveness of FEMA hazard mitigation grants (Rose et al., 2007).

Zobel (2010) extends the characterization of resilience given by Bruneau et al. (2003), and quantifies “triangle–resilience,” a measure of the rapidity with which a system returns to normal functioning, and the robustness (resistance to losing services). This work provides methods for visualizing the trade–off between these two goals of resilience, and has been lately extended for multiple events (Zobel and Khansa, 2011) and preferences of decision makers (Zobel, 2011).

Many of the resilience metrics designed for infrastructure systems other than for road networks are low–resolution models and are not appropriate, or at least not extended, to measuring how component level improvements in a particular community can improve CIS resilience in that community. The metrics developed in Section 2.2.4 are intended to be applicable to any CIS or part thereof, whether a road network or other infrastructure system in the broader CIS.
2.1.2 Interdependent Infrastructure Modeling

Much effort in CIS studies has focused on the modeling of cascading failures due to complex interdependencies among infrastructures. For instance, Rinaldi (2004); Rinaldi et al. (2001) provide conceptual frameworks for understanding interdependent infrastructures as complex, adaptive systems. Multiple dimensions characterize infrastructure interdependencies with diverse pathways that can propagate failure through a system.

Initially, agent–based modeling was a widely used tool for simulating interdependent infrastructure systems, where each infrastructure system corresponds to a particular agent, and interactions among agents are dictated by interdependencies Amin (2000); Rinaldi et al. (2001). Each agent chooses its actions and what it outputs to other agents (infrastructure systems) based on what other agents give. Fuller surveys of these types of models can be found in Rigole and Deconinck (2006) and Pederson et al. (2006). Another summary of many of these types of investigations and more simulation–based models is given by Bagheri and Ghorbani (2007).

Bagheri et al. (2007) introduced an agent–based simulation software architecture where the interactions between agents are based on the services that agents supply one another, a viewpoint of interdependencies adopted in the current paper. Scenario analysis and visualization modules provided the ability to consider infrastructure performance through “what–if” scenarios.

Zio and Sansavini (2011) presented a dynamic simulation model for determining which interdependencies most affect the performance of an individual infrastructures system during a hazard event. The abstract model is mostly intended as a simulation of an infrastructure system where there is the idea of a “load” that causes a component to fail if a threshold is exceeded. Then the load is redistributed over other components in the system. Buldyrev et al. (2010) present another graph theoretic approach. Given two random networks that are dependent on each other, a portion of nodes were removed from each network to determine the cascading effects on the other network. The resulting largest functional subset of the original system is used to calculate system resilience and performance.

Several studies use Leontief Input–Output models to represent the behavior of economic sectors
when subjected to disruptions that can cascade throughout the system. Input–Output Inoperability (IIO) models were developed by Haimes and Jiang (2001) and have been applied to measure economic losses from a deliberate attack against sectors of the economy (Haimes et al., 2005a,b). These models provide a high–level view of the CIS but can be problematic if used to model the CIS in a specific region or community where the precise multiplier effects on the performance of various components is not necessarily known. Additionally, these types of models have not been applied to any optimization frameworks for pre–hazard improvement of CIS components.

Most of these models assume that the CIS components are highly connected, with complex and unknown interdependencies. In reality, complex cascading failures with unexpected far–ranging effects may be rare. Van Eeten et al. (2011) have found little evidence from media accounts of infrastructure for frequent complex cascading failures through an interdependent infrastructure system. Instead, the data indicate cascading failures are generally one–directional failures across well–known connections between infrastructure systems, especially from energy and telecommunications infrastructure systems. The study also concluded that cascading failures are more frequent than generally thought. Based on their results, this paper takes the stance that it is important for a CIS model to be able to incorporate the possibility of cascading failures.

2.1.3 Decision Support Models

Past work in CIS modeling primarily dealt solely with the performance of interdependent infrastructure systems. In recent years, focus has moved to not merely create a simulacrum to gain insight about how the systems function, but also actually identify ways to improve the existing resilience of various parts of the CIS against hazards.

The most common method for improving CIS resilience is through post–hazard restoration and recovery optimization, with a resilient system defined as the one with the lowest recovery costs or fastest time for recovery. Lee et al. (2007) presented a network flow approach to determine the lowest cost recovery given a specific hazard. Decisions consider where to place temporary linkages to support necessary services. A CIS component may have demand for particular levels of flow for com-
modities from other components, without which the component cannot fully supply its lifeline services. This mechanism is used to capture interdependencies among CIS components. The model in Cavdaroglu et al. (2011) extends the earlier work by including scheduling of the restoration decisions.

An earlier study considering interdependencies and optimization of a particular metric, namely, recovery time, came from Xu et al. (2007). Here Markov and network flow models represent the state of links in a CIS consisting of natural gas and electrical systems. Resilience is measured by the restoration time of the system after a disruption. A genetic algorithm optimizes this measure by choosing the amount of funds to invest in each link, where investments would increase the probability a link capacity transitions to state with an improved capacity after a disruption reduces capacity.

While the above decision support models include interdependencies, most optimization models for improving resilience of CIS subject to hazards and disruptions consider a single infrastructure system. Frequently, transportation, or a subsystem of the transportation system, such as bridges, roads, or rail, is examined. Karlaftis et al. (2007) provide a model for allocating funds to improve and repair bridges damaged by natural hazards such as earthquakes. Also, stochastic programming has been applied to the problem of bridge protection by Fan and Liu (2008); Fan et al. (2010), with improvement decisions in the first stage and recourse decisions that represent the state of the transportation system after a hazard.

An abstraction of a transportation network was used in Miller-Hooks et al. (2012) for improving resilience through post–disaster recovery options and pre–disaster preparations. The resource allocation decisions prepare for and recover from an event, through an optimization model formulated as a two–stage stochastic programming problem, whose solution uses the integer L–shaped method (Laporte and Louveaux, 1993).

Peeta et al. (2010) provided one of the few decision models to explicitly include pre–hazard improvements for the resilience of an infrastructure system (particularly a highway) against hazards. The decision–dependent stochastic program uses binary decision variables to decide whether to improve each link in the road network. Improvements reduce the probability that each road link fails, thus improving the possibility that there is a path available for each OD pair. The problem is solved
with a linear approximation and Monte Carlo sampling of the full formulation.

Overall, most of these decision models are either focused on a single infrastructure system, or do not consider what component level decisions should be made to improve performance and resilience of the target system. Additionally, while excellent models have been created for optimizing the recovery from a natural hazard, there has been comparatively less focus on optimizing CIS resilience with pre–hazard engineering improvements for specific components. The work presented in this paper is intended to provide a framework for measuring and improving the most salient life-line services provided by a CIS.

The CIS modeling framework presented in this paper provides metrics for measuring the resilience and performance of a CIS system subjected to a range of hazard scenarios. Rather than consider performance in a given hazard scenario, the focus is on quantifying the resilience of the CIS against multiple possible hazards. Interdependencies among different parts of the CIS are explicitly incorporated, so that the performance of the CIS and the effects of improvements can be viewed on a system–wide basis. These metrics are developed to facilitate applicability to a range of CIS configurations, both as metrics and when incorporated into optimization models for improving system–wide resilience.

2.2 Civil Infrastructure System Modeling Framework

The modeling framework that has been developed consists of six major concepts, depicted in Figure 2.1. Each concept may have different implementations and realizations depending on the critical aspects of resilience examined and the configurations of the CIS. The first four are developed in this section, but engineering improvements and the optimization model are described later in Section 2.4.
2.2.1 System Representation

First, there is the system representation, which informs other concepts in the modeling framework. Within each system representation, the components of the CIS must be designated. Let $C$ be the set of CIS components, indexed by $c$. Components may be road segments, levees, electrical transformers, hospitals, etc. This paper assumes that the CIS must be capable of supporting potential usage of its components for certain services by a population. To represent a component’s contribution to resilience against hazards requires levels of service a component provides after the system has been subjected to a hazard. Define service level of a component as the quantity of service a component provides. For specific components, service level is instantiated as different quantities. It can represent the proportion of the normal vehicles/hour a road link can accommodate (percentage), kilowatts transitted by a transmission line (absolute quantity), or whether or not a wastewater treatment plant is still functioning (binary).

Define $K$ to be the set of lifeline services types that are used in the CIS (e.g., electricity, wastewater, food shipments). Let $s_{h,c,k}$ be the current service level provided by component $c$ of service type $k$ in scenario $h$. Let $\sigma_{c,k}$ be the maximum service level of service type $k$ for component $c$, which may be expressed as 100% or some quantity such as capacity of a road segment in vehicles/hour. In this paper, $s_{h,c,k}$ is considered fixed for each hazard scenario to support the later implementations of metrics and optimization models, though the modeling framework does not require this. For the
metrics below, if it is desired to measure $s_{h,c,k}$ as a percentage, then it is assumed that $s_{h,c,k}/\sigma_{c,k}$ is substituted for $s_{h,c,k}$.

Finally, to fully represent the CIS, the relationships between components must be considered. Components are connected by directed arcs. For components $c_1, c_2$, if there is a link $(c_1, c_2)$ that terminates at $c_2$, then $c_1$ is an *upstream* component of $c_2$; similarly, $c_2$ is *downstream* of $c_1$. Links are divided into two types: transport links and service links. If a link between components can be used by members of a population to travel, then that link is a transport link. Examples include relationships such as which segments of a road network are connected or the layout of train tracks. Service links connect components if the level of a service type provided by a component influences the service levels of another component, i.e., they are interdependent. These linkages may connect a hospital to the electrical grid, or represent the necessity for a levee to protect components of a road network from flooding.

To connect components in the network, sets $TD_c, TU_c, SD_c,$ and $SU_c$ are defined for all $c \in C$. The sets $TD_c$ and $TU_c$ consist of all $c' \in C$ such that $c'$ is respectively downstream or upstream of component $c$ via transport links. For a two–way road segment, adjacent components (i.e., connected road segments) would be both upstream and downstream. The sets $SD_c$ and $SU_c$ consist of the components that are dependent on $c$ and on which $c$ is dependent, respectively.

### 2.2.2 Hazard Scenarios

The second concept of the CIS modeling framework is the hazard scenarios. This concept dictates how hazard scenarios are generated and affect the system representation. Let $H$ be the set of hazard scenarios, indexed by $h$. The effects of hazard scenarios on the CIS may be deterministic, or probabilistic, requiring multiple simulations to estimate the probability a component has of failing and the expected service level after a failure. In this paper each hazard scenario has a deterministic effect on system components, and the probability a scenario occurs is fixed for scenario $h$ by $\pi_h$. Other implementations and realizations of the modeling framework may choose to have the probability of a scenario unfixed, and instead associate failure probabilities directly with components. The imple-
mentations in this paper would be complicated by this type of approach.

CIS performance can also be considered over single or multiple hazard events. Single–event means that a metric may have a range of possible future hazards scenarios, but performance of the infrastructure will be measured after only one of the scenarios occurs. Measuring single–event resilience may still require aggregating performance over the range of scenarios, \( H \). The other option is a multi–event instance of a metric, where a series of hazard events occur, each of which is sampled from a range of possible future hazards. Multi–event performance could be aggregated over a series of sets of hazard scenarios, \( H_1, H_2, \ldots, H_{\epsilon} \), where \( \epsilon \) is the desired number of events.

Multi–event metrics are not addressed in this paper, but a possible strategy to make a metric multi–event is to use an approach like that of the recourse function in stochastic programming ((Birge and Louveaux, 1997)).

2.2.3 Component States

The third concept, component states, dictates how hazard scenarios, engineering improvements and the system representation are combined to form input for the resilience metrics, which will be discussed later. For instance, specific CIS models may require that the service level of a component \( c \) be a function of the service levels in \( SU_c \) (defined as \( s_{h,c,k}(SU_c) \)). The framework has been designed to accommodate this possibility, though explicit dependence of \( s_{h,c,k} \) on the services of the components in \( SU_c \) is omitted generally for clarity. A simple method to constrain service levels based on interdependencies is presented in Section 2.4.2. Additionally, the effects of engineering improvements presented in Section 2.4.1 show another possible realization of the synthesis of system representation, hazard scenarios, and engineering improvements.

2.2.4 Resilience Metrics

The calculation of resilience metrics depends heavily on what instantiations are made of the other input concepts. For instance, system representation used can dictate whether available lifeline ser-
Table 2.1: Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>components of CIS, indexed by $c$</td>
</tr>
<tr>
<td>$H$</td>
<td>hazard scenarios, indexed by $h$</td>
</tr>
<tr>
<td>$K$</td>
<td>usage type, indexed by $k$</td>
</tr>
<tr>
<td>$\sigma_{c,k}$</td>
<td>base (maximum) service level of usage $k$ provided by component $c$</td>
</tr>
<tr>
<td>$s_{h,c,k}$</td>
<td>service usage $k$ currently provided by component $c$ in scenario $h$</td>
</tr>
<tr>
<td>$\pi_h$</td>
<td>probability of scenario $h$</td>
</tr>
<tr>
<td>$TU_c$</td>
<td>set of components with transportation arcs to component $c$</td>
</tr>
<tr>
<td>$TD_c$</td>
<td>set of components with transportation arcs from component $c$</td>
</tr>
<tr>
<td>$SU_c$</td>
<td>set of components with service arcs to component $c$</td>
</tr>
<tr>
<td>$SD_c$</td>
<td>set of components with service arcs from component $c$</td>
</tr>
<tr>
<td>$F$</td>
<td>set of components being measured ($F \subset C$)</td>
</tr>
<tr>
<td>$T$</td>
<td>set of time periods, indexed by $t$</td>
</tr>
<tr>
<td>$\sigma^*$</td>
<td>target performance</td>
</tr>
<tr>
<td>$t_h$</td>
<td>recovery time in scenario $h$</td>
</tr>
<tr>
<td>$R$</td>
<td>set of recovery times, $t_h, \forall h \in H$</td>
</tr>
<tr>
<td>$RT$</td>
<td>aggregate recovery time</td>
</tr>
<tr>
<td>$d_{c,k}$</td>
<td>service type $k$ supplied or demanded at component $c$</td>
</tr>
<tr>
<td>$f_{c,c',k}$</td>
<td>potential usage amount chosen at component $c$ for service $k$ taken from component $c'$</td>
</tr>
</tbody>
</table>

Serviceability

The first metric defined considers lifeline service usage through calculating available service level in a CIS. The serviceability metric, as defined by Berdica (2002), for a component in a road network measures the “possibility to use that link/route/road network.” That definition is extended here beyond

services are considered, or if there are mechanisms to represent a particular pattern of potential usage. The latter would indicate if a population can use services of the CIS as it would prefer, for instance, if drinking water reaches a hospital as opposed to determining the total gallons of available water.
road networks to apply to the possibility to use any component or subsystem. Additionally, service-
ability of the CIS is formally quantified as the expected services provided in each scenario. Any pos-
sible service level, binary or non–binary, is acceptable, as is single–event or multi–event complexity.
As noted, the serviceability metric calculates the amount of available service level in the CIS, and
cannot represent detailed potential lifeline service usage.

As the effects of hazard scenarios are being treated as deterministic, the following method can
be used to calculate serviceability. If \( F \subseteq C \) is the set of components whose serviceability is to be measured, then define the unweighted serviceability in a particular scenario \( h \) as

\[
\bar{s}_h = \frac{1}{|F|} \sum_{c \in F, k \in K} \frac{s_{h,c,k}}{\sigma_{c,k}}, \tag{2.1}
\]

which is the average serviceability of each component. A weighted serviceability can be defined in a
variety of ways. An example is

\[
s_h = \frac{1}{|F|} \sum_{c \in F, k \in K} \frac{s_{h,c,k}}{\sigma_{c,k}}, \tag{2.2}
\]

which allows components with greater service to contribute more to the CIS serviceability. Using
Equation (2.1), the serviceability for an entire CIS is given as

\[
s_{CIS} = \sum_{h \in H} \pi_h s_h. \tag{2.3}
\]

If serviceability is desired for a specific subsystem, then the summation Equation (2.1) could be over
a subset of \( C \).

**Demand Satisfaction**

The *demand satisfaction* metric measures potential lifeline service usage by quantifying the services
a population requires from a particular component. Then the amount of those services a population
can use when the CIS has degraded functionality is calculated. Instances of this metric could be a
measurement of the ability of a road system to support essential transportation, such as fuel or food, or the availability of drinking water to a community.

For a component $c$, $d_{c,k}$ is the amount of service type $k$ demanded at component $c$. When $d_{c,k}$ is negative this indicates that component $c$ can supply the particular service type. The notation can be extended to allow different demands to be required or available in each scenario, by $d_{h,c,k}$. The metric is calculated by solving a multicommodity network flow problem using physical and interdependent connections to calculate the demand satisfied for a chosen group of components. For instance, this metric may be used to ensure that sufficient food and medical supplies are available at a hospital.

A mathematical programming model for solving this problem is given in Equation (2.4), which assumes that the service level of components has already been calculated using interdependencies, and the components whose services are being measured are important for the transport links rather than service links.

\[
\begin{align*}
\min & \sum_{h \in H} \sum_{c \in F} \pi_h \left( d_{c,k} - \sum_{c' \in TU_c} f_{h,c',c,k} \right) \\
\text{subject to} & \sum_{c' \in TU_c} f_{h,c',c,k} - \sum_{c' \in TD_c} f_{h,c',c,k} \geq d_{c,k} \quad \forall c \in C, k \in K, \forall h \in H \\
& \sum_{c' \in TD_c} f_{h,c',c,k} \leq s_{h,c,k} \quad \forall c \in C, k \in K, \forall h \in H \\
& f_{h,c',c',k} \geq 0 \quad \forall h \in H, \forall c \in C, c' \in TD_c, k \in K
\end{align*}
\] (2.4a)

The optimal solution to this problem produces the unsatisfied demand. Obviously, the constant terms in the objective function could be dropped, and the problem could be converted to a maximization problem. This metric is flexible for use in a variety of situations; for instance, to measure the ability of a road network to support fuel supply for gas stations, this model reduces to a single commodity network flow problem. At the other extreme, if the typical non-disrupted usage of the network is modeled with origin–destination pairs for each potential trip, then each trip could be treated as a service type. An additional constraint would be added for mutual capacity on all the road
segments in this case.

**Recovery Time**

The *recovery time* metric measures the number of time periods until the CIS returns to a particular threshold of performance after a hazard. It is not an optimization of the recovery process, but an approximation of how many time periods until the CIS reaches that threshold. The exact length of a time period is not defined here as each hazard and implementation may be best served by different notions. Similarly, the recovery pattern will vary with the types of infrastructure systems involved. In the case of debris and flood clearance from a road network, restoring a road segment may require a serviceable path through the network from clearance equipment to the failed segment along physical arcs.

Let $T$ be the (potentially infinite) set of time periods for recovery, indexed by $t$. It is assumed that at time $t = 0$ a hazard has occurred and affected the performance of the CIS. Let $\sigma^*$ be the target performance for the CIS. Define $t_h$ as the recovery time in scenario $h$, and let the set $R$ be given as $R = \{t_h | h \in H\}$, the set of recovery times for each scenario. The calculation outline is given in Figure 2.2. It is also possible to take the worst recovery time or another quantity as the aggregate recovery time, $RT$.

An important section of the algorithm is left undefined: the determination of which components are restored in each time period, which occurs in Line 7 in the pseudocode of Figure 2.2. This is due to the highly problem-specific nature of this calculation. Depending on the CIS, this could be the result of simulations to estimate when particular road segments are not flooded or output from recovery optimization models, such as those described in Section 2.1.3.

### 2.3 Numeric Experiments for Metrics

The numeric experiments are conducted on an illustrative example based on a real infrastructure system from a coastal city community, whose fictional adaptation is named *Coastal City* (shown in Figure 2.3). This city faces annual possibilities of natural hazards in the form of hurricanes. The
Coastal City CIS has 1,455 road components in the transportation network and 106 components in the levee that protect them.

Each metric was applied to the road network for a range of storm scenarios. In each scenario, it was assumed that a given set of levee segments has failed, which caused flooding in various parts of the road infrastructure. An example scenario showing failed components is given in Figure 2.4. For these examples, 30 hazard scenarios were generated, in addition to a single normal scenario that had no component failure and thus supports full service. The probability of the normal scenario was set to 0.20, while the probabilities for the other scenarios were randomly generated to represent different storm frequencies.

The 30 hazard scenarios were generated by choosing separately for both the levee and road networks the severity and locations for failure. Once the location was chosen, the percentage of nearby components that failed would be dependent on the severity of the hazard. Thus, the hazard scenarios were divided into 5 types, based on where levee failures occurred, and the amount of road network damage occurring directly from the hazard.

For this example, five metrics were chosen to represent the capacity of the road system to provide transportation services: serviceability (weighted and unweighted, $W_{Serv}$ and $U_{Serv}$ respectively);
serviceability restoration time (ServRT); demand satisfaction (DS); and demand satisfaction restoration time (DSRT). For all of these metrics, the level of service as defined in the Highway Capacity Manual Board (2010) was calculated for level of service (LOS) “D”. The quantity given by LOS for a road link represents the average hourly traffic that can be well-accommodated on that road link road. (Note that level of service for road capacity is distinct and separate from service level which is defined in this paper for any CIS component. “LOS” will always refer to the road system capacity notion). The number given by calculating LOS “D” was used as the base service level for each component in the road network to approximate the capacity of a road segment. Supply and demand data were derived from trip generation rates based on 2000 census data and trip generation formulas from a Transportation Research Board Report NCH (1998). Trips (demand and supply) were randomly assigned from census blocks to nearby road components. Fifteen categories grouped each trip into different types for use as data for a multicommodity network flow problem for the demand satisfaction metric.

The serviceability metric was calculated for both weighted serviceability, i.e., four-lane roads are more important than two-lane roads, and unweighted serviceability, where all roads contribute equally to serviceability regardless of individual capacity. For demand satisfaction metrics, the vehi-
Figure 2.4: Locations of Failed Components in Different Hazard Scenarios
cle flow out of each road component was limited by the amount of service level currently available in that scenario for that component. There were 1433 total trips in all categories.

Recovery time was calculated using $\sigma^* = 1$ for serviceability and $\sigma^* = 1433$ for the demand satisfaction metric. Both of metrics were tracked during the recovery process until the CIS had serviceability of $\sigma^*$. Thus, weighted or unweighted serviceability was unimportant in this instance of the metric. Two components of the road network, in the far northwest and northeast parts of the Coastal City region, were designated as recovery bases from where the recovery crews begin restoration. Each base had 100 crews, and each crew was capable of restoring a single reachable and failed component every time period. From each recovery base a random set of failed and reachable components (less than the number of crews) were restored in each time period. When bases have overlapping reachable road components, the bases were combined and crew resources were pooled.

The metric calculations were implemented in C++ and compiled using g++ 4.6.1. The demand satisfaction multicommodity flow problem was solved using CPLEX 12.4. The computer used for solving these problems ran Ubuntu 11.10 with eight cores at 2.8GHz and 8 GB of RAM.

Expected serviceability for this CIS was 66.28%, and weighted serviceability was 71.53%. The expected demand satisfaction was 733.5 trips in all hazard scenarios. The expected recovery time was 62.1 time periods for serviceability and 61.6 for the demand satisfaction metric. A summary of results by scenario is shown in Table 2.2. Figure 2.5 shows the order in which recovery took place for a particular hazard scenario.

Figure 2.6 shows the color-coded correlation matrix between all the metrics and the number of failed components in each scenario. Red indicates a stronger negative correlation coefficient, blue indicates that there is a stronger positive correlation coefficient. For instance, the number of failed components in a scenario as a strong negative correlation with the weighted serviceability and serviceability metrics. The dark blue rectangle in the lower-left of the figure shows that the basic serviceability, weighted serviceability, and demand satisfaction metrics are strongly positively correlated. This indicates that demand satisfaction in the Coastal City road network with the generated hazard scenarios has a strong relationship with serviceability, which measures the available capacity of the
Table 2.2: Serviceability by Scenario. See text for abbreviation descriptions

<table>
<thead>
<tr>
<th>Scenario</th>
<th>WServ</th>
<th>UServ</th>
<th>ServRT</th>
<th>DS</th>
<th>DSRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100%</td>
<td>1.00</td>
<td>0.00</td>
<td>1433.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>73%</td>
<td>67%</td>
<td>77.00</td>
<td>664.00</td>
<td>70.00</td>
</tr>
<tr>
<td>3</td>
<td>82%</td>
<td>78%</td>
<td>83.00</td>
<td>897.00</td>
<td>73.00</td>
</tr>
<tr>
<td>4</td>
<td>77%</td>
<td>73%</td>
<td>39.00</td>
<td>695.00</td>
<td>27.00</td>
</tr>
<tr>
<td>5</td>
<td>68%</td>
<td>60%</td>
<td>82.00</td>
<td>582.00</td>
<td>80.00</td>
</tr>
<tr>
<td>6</td>
<td>57%</td>
<td>47%</td>
<td>103.00</td>
<td>357.00</td>
<td>111.00</td>
</tr>
<tr>
<td>7</td>
<td>61%</td>
<td>52%</td>
<td>93.00</td>
<td>482.00</td>
<td>75.00</td>
</tr>
<tr>
<td>8</td>
<td>89%</td>
<td>86%</td>
<td>25.00</td>
<td>1093.00</td>
<td>20.00</td>
</tr>
<tr>
<td>9</td>
<td>87%</td>
<td>84%</td>
<td>34.00</td>
<td>1082.00</td>
<td>25.00</td>
</tr>
<tr>
<td>10</td>
<td>83%</td>
<td>79%</td>
<td>48.00</td>
<td>909.00</td>
<td>57.00</td>
</tr>
<tr>
<td>11</td>
<td>63%</td>
<td>54%</td>
<td>56.00</td>
<td>621.00</td>
<td>58.00</td>
</tr>
<tr>
<td>12</td>
<td>77%</td>
<td>72%</td>
<td>96.00</td>
<td>706.00</td>
<td>91.00</td>
</tr>
<tr>
<td>13</td>
<td>75%</td>
<td>70%</td>
<td>50.00</td>
<td>627.00</td>
<td>51.00</td>
</tr>
<tr>
<td>14</td>
<td>85%</td>
<td>82%</td>
<td>23.00</td>
<td>918.00</td>
<td>25.00</td>
</tr>
<tr>
<td>15</td>
<td>71%</td>
<td>65%</td>
<td>64.00</td>
<td>621.00</td>
<td>56.00</td>
</tr>
<tr>
<td>16</td>
<td>73%</td>
<td>71%</td>
<td>54.00</td>
<td>813.00</td>
<td>73.00</td>
</tr>
<tr>
<td>17</td>
<td>78%</td>
<td>73%</td>
<td>39.00</td>
<td>693.00</td>
<td>38.00</td>
</tr>
<tr>
<td>18</td>
<td>79%</td>
<td>74%</td>
<td>43.00</td>
<td>861.00</td>
<td>32.00</td>
</tr>
<tr>
<td>19</td>
<td>67%</td>
<td>62%</td>
<td>34.00</td>
<td>575.00</td>
<td>37.00</td>
</tr>
<tr>
<td>20</td>
<td>70%</td>
<td>65%</td>
<td>48.00</td>
<td>663.00</td>
<td>42.00</td>
</tr>
<tr>
<td>21</td>
<td>74%</td>
<td>69%</td>
<td>37.00</td>
<td>638.00</td>
<td>41.00</td>
</tr>
<tr>
<td>22</td>
<td>70%</td>
<td>65%</td>
<td>126.00</td>
<td>620.00</td>
<td>123.00</td>
</tr>
<tr>
<td>23</td>
<td>56%</td>
<td>47%</td>
<td>118.00</td>
<td>393.00</td>
<td>108.00</td>
</tr>
<tr>
<td>24</td>
<td>43%</td>
<td>29%</td>
<td>121.00</td>
<td>177.00</td>
<td>129.00</td>
</tr>
<tr>
<td>25</td>
<td>36%</td>
<td>26%</td>
<td>137.00</td>
<td>207.00</td>
<td>122.00</td>
</tr>
<tr>
<td>26</td>
<td>58%</td>
<td>49%</td>
<td>60.00</td>
<td>417.00</td>
<td>63.00</td>
</tr>
<tr>
<td>27</td>
<td>56%</td>
<td>46%</td>
<td>110.00</td>
<td>418.00</td>
<td>97.00</td>
</tr>
<tr>
<td>28</td>
<td>47%</td>
<td>40%</td>
<td>76.00</td>
<td>266.00</td>
<td>78.00</td>
</tr>
<tr>
<td>29</td>
<td>48%</td>
<td>40%</td>
<td>66.00</td>
<td>198.00</td>
<td>71.00</td>
</tr>
<tr>
<td>30</td>
<td>35%</td>
<td>27%</td>
<td>108.00</td>
<td>139.00</td>
<td>127.00</td>
</tr>
<tr>
<td>31</td>
<td>36%</td>
<td>28%</td>
<td>149.00</td>
<td>182.00</td>
<td>146.00</td>
</tr>
</tbody>
</table>
Figure 2.5: Recovery operations for a particular scenario. The highlighted color of a link indicates the time period the link was restored. Total restoration time was 124 periods.

Figure 2.7 compares metric performances for each hazard scenario. Again, the lower-right area shows strong positive correlations between demand satisfaction, serviceability and weighted serviceability. In fact, most of the values plotted have strong correlations, whether positive or negative.

There are different possible explanations for this. One is that this example is simple enough that a broad metric such as serviceability captures most of the aspects of resilience for the CIS. Another interpretation that could be made is that resilience in CIS does not require complicated metrics, but that various aspects can be measured by a straightforward and easy-to-understand function, such as serviceability or even the number of failed components.
Figure 2.6: Level Plot of Correlation Matrix
Figure 2.7: Scatter Plot of Comparing by Scenario Resilience Metric Performances and Number of Failed Components
2.4 Optimization Models

The framework’s optimization models provide formulations for systematically improving resilience metrics. The optimization models choose what type of improvement decision each CIS component receives according to a given budget. Improvement decisions could be retrofits of existing components, or the construction of a new component better able to withstand hazards. The CIS, with improvements, is then subjected to a range of possible hazard scenarios, with a probability distribution determining the likelihood of each hazard scenario. If a component loses service level in a particular hazard scenario due to interdependencies or directly from the hazards effects, then an improvement decision made for the component can affect the service level of that component in two ways. The improvement could reduce the probability of failure in that scenario, and the improvement could increase the service level that the component provides in that scenario. After improvement effects are determined for each component, constraints ensure that the effects of interdependencies on service levels are respected in each hazard scenario. The objective function aggregates components’ service levels to calculate the expected CIS resilience over the set of hazard scenarios according to the chosen resilience metric.

This section does not provide specific algorithms for solving instances of the framework, as different instances may result in different types of optimization problems. The optimization model instance used for the numeric example adheres to a mixed integer/linear programming formulation. The notation for data used throughout this section is summarized in Table 2.3.

2.4.1 Decision Variables

Improvements options for the CIS components are the decision variables of the optimization models. Improvements to components could include retrofits of existing components, such as raising the height of a road bed, or design decisions, such as building a water treatment plant in a new location. Binary decisions are used for CIS improvement choices to reflect the discreteness of choosing to undertake an improvement project or not. Define $U_c$ as the set of possible improvements for component $c$, which will be indexed by $u$. Let $x_{c,u}$ represent whether improvement $u$ is chosen for
Table 2.3: Notation for the Optimization Models

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_h$</td>
<td>set of components losing performance directly from hazard in scenario $h$.</td>
</tr>
<tr>
<td>$U_c$</td>
<td>upgrades possible at component $c$</td>
</tr>
<tr>
<td>$\pi_{h,c}$</td>
<td>probability of service occurring for component $c$ in scenario $h$</td>
</tr>
<tr>
<td>$\lambda_{h,c,u,k}$</td>
<td>benefit for service usage $k$ provided by component $c$ in scenario $h$ if upgrade $u$ is chosen</td>
</tr>
<tr>
<td>$\mu_{h,c,u,k}$</td>
<td>effect on probability of service usage $k$ provided in scenario $h$ by component $c$ if upgrade $u$ is chosen</td>
</tr>
<tr>
<td>$\eta_{c,c',k}(\cdot)$</td>
<td>function for service usage $k$ available at component $c$ based on service level provided by component $c'$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>improvement budget</td>
</tr>
<tr>
<td>$\kappa_{c,u}$</td>
<td>cost of improving component $c$ with upgrade $u$</td>
</tr>
<tr>
<td>$\phi_{h,c,k}$</td>
<td>base service level of usage $k$ at component $c$ in scenario $h$ (when $c \in A_h$)</td>
</tr>
<tr>
<td>$s_{h,c,k}$</td>
<td>service level achievable for usage $k$ at $c$ in scenario $h$ after improvements</td>
</tr>
<tr>
<td>$p_{h,c,k}$</td>
<td>probability service level in scenario $h$ is achieved for usage $k$ at $c$ after improvements</td>
</tr>
<tr>
<td>$x_{c,u}$</td>
<td>binary decision variable, 1 if improvement $u$ is chosen for component $c$ and 0 if not</td>
</tr>
</tbody>
</table>
component \( c \) such that it is equal to 1 if improvement \( u \) is chosen for \( c \), and 0 if not. In this example, it is assumed that only one improvement may be chosen per component. Thus, if \( x_{c,0} \) is defined as the “do–nothing” option for each component, with cost 0 and no effects, the constraint

\[
\sum_{u \in U_c} x_{c,u} = 1 \quad \forall c \in C
\]  

(2.5)

ensures that each component receives exactly one improvement decision.

It is assumed that an improvement directly affects only the component to which it is associated. This follows the framework goal of aggregating component performance into CIS performance. The assumption does not prevent additional constraints from connecting decisions made for related components if needed. For example, adjacent segments of the road might need to receive the same improvement type. Another consideration is that an improvement may not behave uniformly across all possible hazard scenarios. Instead, a decision that is effective in one scenario may be completely ineffective or even detrimental in another. The optimization framework accommodates scenario-specific effects for each improvement decision.

Improvement decisions can reduce a component’s likelihood of failure and increase its available service level in a hazard scenario where the component would otherwise fail. Given improvement \( x_{c,u} \) is chosen, \( \mu_{h,c,u,k} \in [0,1] \) is the amount of reduction of the probability of service level loss for service usage \( k \) in scenario \( h \) on component \( c \). The increase in service level for service usage \( k \) at component \( c \) in scenario \( h \) if improvement \( x_{c,u} \) is selected is \( \lambda_{h,c,u,k} \).

In this paper, it is assumed that all improvement decisions are made prior to the hazard and all implementations are fully complete by the time the hazards could affect the CIS. This removes uncertainty from the effects of completion status of an improvement project at the time of the hazard.

A critical aspect of resilience optimization is the cost of the proposed plan. Given \( \kappa_{c,u} \) is the cost of improvement \( u \) for component \( c \), the budget constraint is

\[
\sum_{c \in C} \sum_{u \in U_c} \kappa_{c,u} x_{c,u} \leq \beta,
\] 

(2.6)
where \( \beta \) is the available budget. In multi-event instances, the budget \( \beta \) may be stage-wise, to reflect that funding levels for the CIS may change over time. The left side of this constraint could also be used as an objective function to find cost-effective solutions meeting particular target levels of resilience.

### 2.4.2 Dependency Constraints

The dependency constraints govern how the service levels of interdependent components affect one another. For each hazard scenario there are two kinds of “failed” or reduced functioning components: those losing service levels directly from hazard effects and those losing service from cascading failures through interdependencies. Let \( A_h \) be the set of components losing service directly from the effects of hazard scenario \( h \), and the remaining components, \( C \setminus A_h \), potentially lose service level due to a cascading failures in scenario \( h \). Within each hazard scenario, only the latter group of components requires dependency constraints.

It is assumed that the service levels of components in \( A_h \) are dictated by the hazard scenario and not dependencies. Thus, their service level is bounded by the hazard scenario effects and the effects of chosen improvements. The bound is expressed as:

\[
    s_{h,c,k} \leq \phi_{h,c,k} + \sum_{u \in U_c} x_{c,u} \lambda_{c,u,k} \quad \forall k \in K, \forall c \in A_h, \forall h \in H, \tag{2.7}
\]

where the hazard scenario data \( \phi_{h,c,k} \in [0, \sigma_{c,k}] \) gives the service level available for service usage \( k \) at component \( c \) in hazard scenario \( h \). The summation over \( x_{c,u} \) calculates the effect improvements have on the service level of \( s_{h,c,k} \).

For the components \( c \in C \setminus A_h \), the bound on \( s_{h,c,k} \) must be a function of the service levels at service linked upstream components and the effects of the improvements chosen for \( c \). The function \( \eta_{c,c',k}(s_{h,c',k}) \) gives the maximum service level \( c \) can achieve based on the service level of \( c' \in SU_c \). Then, the following constraint bounds the service level component \( c \) can achieve in hazard scenario
The constraint limits the service level of a component, $s_{h,c,k}$, to the service achievable based on its dependencies and the effects of the improvements made for the component. If improvement $x_{c,u}$ is equal to 1, then the minimum service level of $s_{h,c,k}$ is increased by $\lambda_{h,c,k}$. For the numerical experiments, this paper defines

$$\eta_{c,c',k}(s_{h,c,k}) = \frac{s_{h,c',k}}{\sigma_{c',k}}$$  

so that Equation (2.8) becomes

$$s_{h,c,k} \leq \sigma_{c,k} \left( \frac{s_{h,c',k}}{\sigma_{c',k}} \right) + \sum_{u \in U_c} x_{c,u} \lambda_{c,u,k} \quad \forall k \in K, \forall c \in C \setminus A_h, \forall h \in H, \forall c' \in SU_c. \quad (2.10)$$

Equation (2.10) requires that service level is the same percent of full availability as the lowest percent availability of all of its upstream service linked components. Other representations could be used as needed to model the service level of a component based on the service level of the interdependent upstream components.

Formulating individual component probability of failure within each scenario is similar. Let $p_{h,c,k}$ be the probability that the service level of component $c$ in scenario $h$ for usage type $k$ is achieved. For components $c \in A_h$, the base probability of failure is:

$$p_{h,c,k} \geq \pi_{h,c,k} - \sum_{u \in U_c} x_{c,u} \mu_{c,u,k} \quad \forall c \in A_h, \forall h \in H, \forall k \in K. \quad (2.11)$$

Note $\pi_{h,c,k} = 1.0$ for nearly all instances. This means that a component that fails under particular hazard scenario has a probability of 1 of failing if no improvement is chosen for that component. For
components $c \in C \setminus A_h$, the following constraint is needed:

$$p_{h,c,k} \geq \eta_{c,c',k}(p_{h,c',k}) - \sum_{u \in U_c} x_{c,u} \mu_{c,u,k} \quad \forall c \in C \setminus A_h, \forall h \in H, \forall c' \in SU_c, \forall k \in K$$  \hspace{1cm} (2.12)

For a given component $c$, the constraint forces the probability that component $c$ has service level $s_{h,c,k}$ to be at least the (improved) probabilities based on the probabilities that $c' \in SU_c$ has service level $s_{h,c',k}$.

In the optimization models in the numeric experiments, the function $\eta_{c,c',k}(p_{h,c',k})$ for failure probabilities can be defined as $\eta_{c,c',k}(s_{h,c,k}) = p_{h,c',k}$ and Equation (2.12) can be written as:

$$p_{h,c,k} \geq p_{h,c',k} - \sum_{u \in U_c} x_{c,u} \mu_{c,u,k} \quad \forall c \in C \setminus A_h, \forall h \in H, \forall c' \in SU_c, \forall k \in K.$$ \hspace{1cm} (2.13)

This instantiation of $\eta_{c,c',k}$ states that the probability component $c$ has the service level given by the hazard scenario is at least as highest probability of failure among the components that $c$ depends on. Improvements, however, can potentially lower the failure probability. Other instantiations are possible.

### 2.4.3 Objective Function

Each instance of the optimization framework requires a different objective function value depending on the chosen metric. The objective function for weighted serviceability is presented here. The expression will be developed in three parts: serviceability contributed from components in hazard scenarios that degrade CIS performance, serviceability contributed from components whose failure probabilities are reduced in hazard scenarios, and serviceability contributed by components in $h_0$, which is the “nothing happens” hazard scenario.

For hazard scenarios $h \neq h_0$ and for component $c$, the expression $\pi_h p_{h,c,k} \sum_{k \in K} s_{h,c,k}$ gives the serviceability as a product of the probability of failure and the service level achieved. The partial
serviceability from each hazard scenario (where a hazard occurs) is:

\[
\sum_{c \in C} \sum_{k \in K} \sigma_{c,k} \frac{1}{\pi_h} \sum_{\substack{h \in H \atop h \neq h_0}} \sum_{\substack{c \in C \atop \sigma_{c,k} \neq 0}} \sum_{\substack{k \in K \atop \sigma_{c,k} \neq 0}} p_{h,c,k} s_{h,c,k}.
\] (2.14)

This expression assumes that there is probability \( p_{h,c,k} \) that component \( c \) has service level \( s_{h,c,k} \) in scenario \( h \). Next, the serviceability contributed in the complement event (i.e., component \( c \) does not have service level \( s_{h,c,k} \) with probability \( 1 - p_{h,c,k} \)) is given.

It is assumed that the service level of \( c \) in the event that occurs with probability \( 1 - p_{h,c,k} \) in scenario \( h \) is \( \sigma_{c,k} \). Thus, \( \pi_h (1 - p_{h,c,k}) \) is the probability that a component has service level \( \sigma_{c,k} \) in scenario \( h \neq h_0 \). For components with the \( p_{h,c,k} < 1 \) in such hazard scenarios, the serviceability contribution is:

\[
\sum_{c \in C} \sum_{k \in K} \sigma_{c,k} \frac{1}{\pi_h} \sum_{\substack{h \in H \atop h \neq h_0}} \sum_{\substack{c \in C \atop \sigma_{c,k} \neq 0}} (1 - p_{h,c,k}) \sigma_{c,k}.
\] (2.15)

If a component has \( s_{h,c,k} = \sigma_{c,k} \) initially (i.e., does not fail or lose service) and improvements cause \( p_{h,c,k} < 1 \), then Equations (2.15) and (2.14) combine so that component \( c \) contributes the same amount of service level for usage \( k \) as if \( p_{h,c,k} = 1 \).

Equations (2.14) and (2.15) are combined and simplified to express the combined serviceability of hazard scenarios \( h \neq h_0 \) as:

\[
\sum_{c \in C} \sum_{k \in K} \sigma_{c,k} \frac{1}{\pi_h} \sum_{\substack{h \in H \atop h \neq h_0}} \sum_{\substack{c \in C \atop \sigma_{c,k} \neq 0}} (1 - p_{h,c,k}) \sigma_{c,k} + p_{h,c,k} s_{h,c,k}.
\] (2.16)

The serviceability contributed by scenario \( h_0 \) to the overall system performance is expressed by:

\[
\sum_{c \in C} \sum_{k \in K} \sigma_{c,k} \frac{1}{\pi_{h_0}} \sum_{\substack{c \in C \atop \sigma_{c,k} \neq 0}} \sum_{\substack{k \in K \atop \sigma_{c,k} \neq 0}} \sigma_{c,k}.
\] (2.17)

Combining Equations (2.16) and (2.17), the following expression for the weighted serviceability is
created:

$$\frac{1}{\sum_{c \in C} \sum_{k \in K} \sigma_{c,k}} \left( \sum_{c \in C} \sum_{k \in K} \pi_{h_0} \sigma_{c,k} + \sum_{h \in H} \sum_{k \in K} \sigma_{c,k} \sigma_{c',k} (1 - p_{h,c,k}) + p_{h,c,k} s_{h,c,k} \right).$$  \hspace{1cm} (2.18)

This expression for weighted serviceability is convertible to unweighted serviceability by replacing the fraction $1/(\sum_{c \in C} \sum_{k \in K} \sigma_{c,k})$ with $1/|C|$, replacing each $\sigma_{c,k}$ term with 1 and dividing each $s_{h,c,k}$ term by $\sigma_{c,k}$.

### 2.4.4 Serviceability Optimization Model

A general instance of a serviceability optimization model is presented here as a mathematical programming problem. This model selects improvements that affect either service levels after a hazard occurs or the probability that a component loses service level in a hazard scenario. Dependency and failure constraints from Section 2.4.2 are ensure that interdependencies are respected.

$$\text{maximize} \quad \frac{1}{|C|} \left( \sum_{c \in C} \sum_{k \in K} \pi_{h_0} \sigma_{c,k} + \sum_{k \in K} \sigma_{c,k} \sigma_{c',k} \sum_{h \in H} \pi_{h_0} (1 - p_{h,c,k}) + p_{h,c,k} s_{h,c,k} \right)$$

(2.19a)

subject to

$$s_{h,c,k} \leq \phi_{h,c,k} + \sum_{u \in U_c} x_{c,u} \lambda_{c,u,k} \quad \forall k \in K, \forall c \in A_h, \forall h \in H$$

(2.19b)

$$s_{h,c,k} \leq \eta_{c,c',k}(s_{h,c',k}) + \sum_{u \in U_c} x_{c,u} \lambda_{c,u,k} \quad \forall h \in H, \forall c \in C \setminus A_h, \forall c' \in SU_c, \forall k \in K$$

(2.19c)

$$p_{h,c,k} \geq \pi_{h,c,k} - \sum_{u \in U_c} x_{c,u} \mu_{c,u,k} \quad \forall h \in H, \forall c \in A_h, \forall k \in K$$

(2.19d)

$$p_{h,c,k} \geq \eta_{c,c',k}(p_{h,c',k}) - \sum_{u \in U_c} x_{c,u} \lambda_{c,u,k} \quad \forall h \in H, \forall c \in C \setminus A_h, \forall c' \in SU_c, \forall k \in K$$

(2.19e)

$$\sum_{c \in C} \sum_{u \in U_c} \kappa_{c,u} x_{c,u} \leq \beta$$

(2.19f)

$$\sum_{u \in U_c} x_{c,u} = 1 \quad \forall c \in C$$

(2.19g)
The objective function, Equation (2.19a), calculates unweighted serviceability for all CIS components based on the effects of the chosen improvements, $x_{c,u}$. Note that this objective function is quadratic due to the $p_{h,c,k}$ terms, but $\mu_{h,c,k}$ or $\lambda_{h,c,k}$ may be fixed to 0 for all $c \in C$, $h \in H$ and $k \in K$ so the model instance simplifies to a mixed integer–linear programming model. These simplifications are additional assumptions for improvement decisions, that the effects of improvements can only reduce failure probabilities or increase service level, but not both.

The model was verified on data generated for small example with four road components, three levee components, three hazard scenarios (including the “nothing happens” scenario), and one type of service usage in the set $K$. In this example, each levee component protected a different subset of road components from flooding. The service levels of each component in each hazard scenario were verified for each possible combination of improvements to ensure that component improvements produced expected results. Probabilities of failure for each component in each scenario were verified for selected improvement plans. Each improvement plan was verified assuming either $\mu_{h,c,k} = 0$ or $\lambda_{h,c,k} = 0$ for all $c \in C$, $h \in H$, and $k \in K$.

### 2.5 Numeric Experiments for Optimization Models

This section presents the results of a case study based in Coastal City for optimizing the resilience of the road network. This example uses a CIS and hazard scenarios that are identical to that presented in Section 2.3. The serviceability resilience metric was chosen to measure the capacity of the road system (as in Section 2.3). Serviceability was chosen for two reasons. First, the results of the numeric examples for the metrics indicated that serviceability alone may be sufficient for capturing resilience of this CIS under the generated hazard scenarios. Second, with certain assumptions, the optimiza-
tion model for serviceability is simple enough to be solved by a general linear/integer programming solver.

Three optimization model instances were solved. The first model, $M_1$, used unweighted serviceability (Equation (2.19a)) as its objective function, and the second model ($M_2$) used weighted serviceability (Equation (2.18)). Both models used the constraints in Equations (2.19b)–(2.19j). To remove the nonlinear terms in the objective functions, it is assumed that improvement decisions only increase service levels, but do not reduce the probabilities of failure. This was represented by setting $\mu_{h,c,k} = 0$ for all $c \in C$, $h \in H$ and $k \in K$, which fixed the $p_{h,c,k}$ terms in the objective functions at 1. The third model ($M_3$) removed the budget constraint (Equation (2.19f)) and fixed all improvement choices to their highest possible level to determine the maximum improvement possible and the cost of that plan. With all improvements fixed, the objective function of $M_3$ was able to be either weighted or unweighted with no difference.

Data for the effects and costs of the engineering improvements were randomly generated. Three levels of improvements were possible for each component. The first level of improvement had no effect and cost nothing, i.e., the “do–nothing” decision. Higher levels had increasingly greater effects on the service level a component provided or the probability a component failed. All improvements costs were generated based on the length of the component, whether a levee component or a road segment. The budget for improvements was set to $3 million, which allowed around 140 improvements combined for road and levee components.

The optimization model instances were implemented in C++, compiled using g++ 4.6.1 and solved using CPLEX 12.4, on a computer running Ubuntu 11.10 with eight cores running at 2.8GHz and 8 GB of RAM. All model instances had 104,911 variables, of which 4,842 were binary, and 170,960 constraints. All model instances were solved in under a few seconds.

A summary of the results is shown in Table 2.4. Initial serviceability for this system is calculated as 66.27% for the unweighted case and 71.51% for the weighted case in Section 2.3. The results in the “Serviceability” column of the table show the final improved objective function. The summary of the types of decisions made for each optimization model in Table 2.4 shows that models $M_1$ and $M_2$
Table 2.4: Optimization Model Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Serviceability</th>
<th>Levee Improvements</th>
<th>Road Improvements</th>
<th>Budget Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type 2</td>
<td>Type 3</td>
<td>Type 2</td>
</tr>
<tr>
<td>$M_1$</td>
<td>69.17%</td>
<td>19</td>
<td>18</td>
<td>119</td>
</tr>
<tr>
<td>$M_2$</td>
<td>73.89%</td>
<td>19</td>
<td>18</td>
<td>113</td>
</tr>
</tbody>
</table>

The improvements chosen for models $M_1$ and $M_2$ are displayed in Figure 2.8. The solution for model $M_1$ in Figure 2.8a shows that improvements are generally made to levees or the roads that are affected by multiple levee failures. Many of the improvements chosen with model $M_2$ (Figure 2.8b) are similar, demonstrating that unweighted and weighted serviceability, as indicated in the numeric experiments with the metrics, tend to agree. The slight difference between the solutions chosen for the two models can be seen on the roads that leave the region to the north and northwest. Both roads are highways and as such have high capacity, so a failure in one of these roads has a more severe impact on the road network (weighted) serviceability. Model $M_2$ accordingly places slightly more emphasis on improving these road components than the solution to $M_1$ does.

In the southeastern region of Coastal City, roads components are dense and each levee failure leads to approximately the same roads flooding. Thus, both models tend to choose improvements for the road components rather than the levee components in this area. In the in the south and southwestern part of Coastal City, levee failures produce fewer overlapping flooded components. The distribution of road and levee improvements for this section of the city was more mixed as a result.

Table 2.5 displays the initial and improved metric values for Coastal City from the three model instances. The two middle columns show the metric results after the model was solved for unweighted and weighted serviceability, respectively. The final column in the model shows the best possible met-
Table 2.5: Resilience metrics after service optimizations (italics indicate a metric that was used as the objective function for the given decision model)

<table>
<thead>
<tr>
<th>Resilience Metric</th>
<th>Initial</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$UServ$</td>
<td>66.27%</td>
<td>69.17%</td>
<td>69.13%</td>
<td>75.30%</td>
</tr>
<tr>
<td>$WServ$</td>
<td>71.51%</td>
<td>73.85%</td>
<td>73.89%</td>
<td>79.02%</td>
</tr>
<tr>
<td>$ServRT$</td>
<td>61.88</td>
<td>59.43</td>
<td>59.06</td>
<td>56.44</td>
</tr>
<tr>
<td>$DS$</td>
<td>733.51</td>
<td>779.83</td>
<td>778.77</td>
<td>869.01</td>
</tr>
<tr>
<td>$DSRT$</td>
<td>61.89</td>
<td>59.51</td>
<td>58.52</td>
<td>55.81</td>
</tr>
</tbody>
</table>

ric results that were possible in Coastal City (determined by model $M_3$). The cost for $M_3$ was $32 million. The large disparity in cost indicates that there could be great cost savings obtained by considering improvements to civil infrastructures on a system–wide basis. The results demonstrate that the presented modeling framework can measure and improve the performance and therefore the resilience of the CIS infrastructure.

2.6 Observations and Future Work

The CIS modeling framework presented fuses different aspects of critical and civil infrastructure modeling. Metrics quantify the system–wide performance and resilience, while explicitly including interdependencies between different infrastructure subsystems. The metrics focus not on how a population uses the CIS, which necessitates the challenge of modeling human behavior, but whether the CIS can support the critical lifeline services necessary for modern society. Moreover, the framework describes a relationship between decisions and hazard scenarios facilitating the insertion of metrics into optimization models to increase future resilience of the CIS against the same natural hazards.

As discussed in the numeric examples, results from this framework are only applicable with reasonable data. General trends about the risk from dependencies, however, and the patterns of improvements given those risks, can be obtained even from only partly real–world based data.
There are two main issues to be addressed with continued research on the framework. First, it may be extremely challenging to obtain all the necessary data for optimization modeling, and for certain decision-makers, even finding necessary data for calculating metrics may be difficult. For instance, the knowledge of the set $SU_c$ may be difficult to come by. The results of the study conducted in Van Eeten et al. (2011) suggest that most interdependencies in the CIS are well-known. Data describing a broad range of hazard scenarios may also be difficult to generate. In this paper, it was generally assumed that numeric simulations can provide enough possible hazard scenarios to give a good estimate of the performance of the CIS against future unknown hazards. In many cases, this is true, but the resources required to run large-scale finite-element models for detailed and accurate scenarios could be prohibitive for certain stakeholders and decision makers. This concern with gathering data through computational resources becomes more crucial when optimization models, and thus the effects of decisions on component level, are desired.

The second main issue in continued development of this framework is also related to the computational resources and issues with scalability. Though the Coastal City example is reasonably-sized, for a larger region, the optimization models could become intractable. For example, optimizing to improve demand satisfaction requires around 100,000 flow variables for each scenario, which for 30 scenarios totals approximately 3 million variables. Methods are available for solving this scale of problem, but generally special implementations are needed.

Despite these issues, the CIS modeling framework presented in this paper provides a basis for optimizing system-wide resilience through pre-hazard improvements. The framework has explicit mechanisms for including interdependencies and even cascading failures due to dependencies. Future algorithmic work can improve the performance of the algorithms even in large-scale environments. Additionally, data dependency may be lessened by developing methods for robust and diverse decision making, giving decision-makers less sensitive but well-performing solutions for consideration.
Figure 2.8: Improvement Decisions for Coastal City

(a) Optimizing (Unweighted) Serviceability ($M_1$)

(b) Optimizing Weighted Serviceability ($M_2$)
Stage–wise Improvements of Civil Infrastructure System Resilience

This paper extends the framework developed in Piper et al. (2013a) (Chapter 2) to consider multi–event resilience to natural hazards. A model is presented and tested on realistic data based on real world civil infrastructure systems. In addition, algorithmic methods are introduced that attempt to address the larger scale required by multi–event optimization models. The effectiveness of the algorithm at addressing a greater degree of uncertainty is assessed.

3.1 Introduction

A civil infrastructure system (CIS) must provide support for critical lifeline services to a population. Pre–hazard improvements to CIS components can reduce or eliminate service losses that hazards can cause, thus lifeline services can continue to be provided. Optimization models can be used to choose where to allocate resources to component improvement to increase system–wide resilience of the infrastructure systems.

Past modeling efforts have neglected some approaches by which the resilience of the CIS in a
region can be enhanced. For instance, budgets for certain CIS may have yearly funding changes. Funding uncertainty can make intensive, long-term projects difficult to plan. Projects may require incremental steps to complete. The number of necessary steps has the potential to expose the CIS to multiple hazard events before all planned improvement tasks have been completed. Additionally, protecting the CIS against multiple events could be strengthened by considering the order in which hazard scenarios may occur and prioritizing improvements accordingly.

A possibility for explicitly hedging against multiple, sequential hazard events is to separately, but sequentially, consider single hazard events. Yet improvement decisions that are made for a single event may become less effective if aspects of the CIS change significantly between events. For instance, population distribution in a region could change drastically, and the CIS may not properly support both population states. Additionally, a less effective improvement decision now may have a stronger effect in future hazard scenarios, in the case of climatic shifts. Improvements should be chosen with regard to their effectiveness over different planning horizons. Sequential optimization of improvement decisions does not guarantee this, nor does it avoid the possibility of having to improve a component already improved in an earlier stage.

This paper presents optimization models that account for multi-event considerations when planning and designing improvements for CIS resilience. The resilience metrics of Piper et al. (2013a) are extended to multi-event hazard scenarios to quantify the resilience of a CIS to multiple hazard scenarios. The general approach used here for modeling multi-event hazard scenarios is a multi-stage stochastic programming problem. A multi-stage stochastic mixed integer programming problem is described, capable of handling interdependencies among components or subsystems and simultaneously optimizing a certain resilience metric across multiple hazard scenarios. The potentially large-scale of the problem necessitates either high-performance computing with a cluster, or a special algorithmic approach to reduce memory requirements to aid in tractable computation of solutions. In this paper, an L-shaped method, based on Benders decomposition, is applied to solve these problems. The difficulty of multi-stage stochastic programming is addressed by formulating the optimization model with block separable recourse, which allows overcoming the multi-stage nature of
the problem by creating a model more amenable to the optimization methods of two–stage stochastic programming problem. Enhancements to Benders decomposition are implemented to attempt to improve solution time.

Numeric examples test this approach on realistic data from a coastal community. Different resilience metrics are optimized and the performance of the Benders decomposition and other approaches are compared. Measurements of the value of the stochastic programming approach and benefits that could be obtained from better prediction of future hazards are quantified.

3.2 Background

Related work in the problem domain can be found in past efforts in CIS modeling, hazard recovery optimization modeling, and fortification models for multicommodity supply and demand networks. An overview of some literature addressing the solution of stochastic programming problems is provided as well.

3.2.1 Interdependent CIS Modeling

Modeling of the CIS has generally focused on two areas: simulations utilizing what–if scenario analysis to measure system performance and recovery after a hazard, and optimization models that determine recovery strategy and actions after a given hazard occurs.

Models for system performance usually fall into a few categories: simulations using agent–based models (e.g., Bagheri and Ghorbani 2007; Bagheri et al. 2007), input–output inoperability models (e.g., Haimes et al. 2005a,b; Santos 2006), and system dynamics simulations (e.g., Buldyrev et al. 2010; Zio and Sansavini 2011). Only a brief review is given of these models in this paper; a fuller review is given in Piper et al. (2013a). These models do not have explicit mechanisms to search for decisions to improve CIS resilience. Instead they generally provide what–if, single–scenario analysis. Still, these models can be used to gain insights into the behavior of interdependencies in an infrastructure system. The remaining review of other investigations focuses on decision models designed to improve
the resilience of a CIS subject to hazards.

Retrofits and improvements to bridges subject to seismic hazards are examined by Karlaftis et al. (2007), where a model is provided for allocating funds to improve and repair bridges. A genetic algorithm is used to optimize the three-stage solution approach, which distributes the funds given a particular hazard scenario has occurred. Lee et al. (2007) described a restoration model that also optimizes restoration in an interdependent infrastructure system given an initial hazard and operating states of the included parts of the CIS. A mixed-integer formulation is used, where interdependencies among infrastructure systems are represented as flows of services on a network. The model can be solved quickly on a reasonably-sized network using commercial solvers. Cavdaroglu et al. (2011) extended the model presented in Lee et al. (2007) by integrating restoration actions (as before) with the scheduling of those actions. Computational difficulties with the extended model necessitate the development and implementation of a heuristic to obtain the solution.

When multiple scenarios are considered for recovery optimization, stochastic programming has been used to model the decisions. Barbarosoğlu and Arda (2004) presented a two-stage stochastic program for distributing first-aid during disaster recovery on a transportation network. The underlying model is a multicommodity, multi-modal network flow with uncertainties in supply, demand and arc capacity. The models were solved for eight earthquake scenarios using GAMS/OSL. The value of the stochastic solution and expected value of perfect information were calculated for each of the scenarios.

Some studies have examined pre-disaster improvements of the system against hazards. These models are usually related to fortification models, a broad class of models intended to locate and protect vulnerable parts of supply chains from disruptions. Snyder et al. (2006) gives a survey of fortification models that address the need for decision-makers to plan for supply chain disruptions. In that work, each model is realized as a mixed integer-linear program, though it is noted that some may be extended to find single solutions to multiple scenarios, thus falling in the domain of two-stage stochastic integer programs. Many fortification models for supply chains focus on interdiction models. For instance, Aksen et al. (2009) considered a system with $p$ facilities where an attacker
will attempt to disrupt \( r \) of them. Fortification decisions are made to minimize the effects of the disruption on the overall system. This problem is a leader–follower Stackelberg game that is modeled using a bilevel integer program. Solutions are obtained for a suite of randomly generated instances, and the tendencies of problem parameters are examined.

A drawback to directly applying interdiction–focused models to CIS resilience modeling is that interdiction often protects specifically against the worst–case scenario, which is generally inappropriate for natural hazards. The first is that the approach frequently assumes an intelligent adversary reduces component performance, which is highly unlikely to be reasonable for natural hazards. Secondly, the worst–case scenarios for natural hazards are often devastating beyond what an intelligent adversary with limited resources can accomplish and little can be done to protect against these cataclysmic events. Using these models for natural hazards modeling could result in effort wasted on averting scenarios that are extremely unlikely.

When improving resilience of the CIS over a range of hazard scenarios, methods that aggregate scenario performance, such as the recourse function of stochastic programming, are frequently appropriate. The related models by Liu et al. (2009) and Fan et al. (2010) use a two–stage stochastic integer program to determine retrofit decisions on bridges in a transportation network to protect against seismic risk. The objective is to minimize the expected costs of traveling on the network after a hazard (travel times are converted to monetary costs and added to retrofit and repair costs). These efforts also assume that retrofitting completely removes the probability of failure from a link. Solutions are obtained by applying an L–shaped method to decompose the problem, allowing exploitation of the multicommodity network flow structure of the subproblems. A case study shows the value of this solution procedure for a problem with six hazard scenarios.

The model developed by Xu et al. (2007) combines recovery optimization and pre–hazard improvements to the CIS. Markov chain and network flow models represent the state of links in a CIS containing gas and electrical systems, along with the interdependencies between these systems. Resilience is measured by the time to recovery, which is affected by investments made for links in and between the the gas and electrical systems. The improvement strategy is optimized using a genetic
algorithm, and the effectiveness of each strategy is calculated by evaluating performance on a sample of possible disruptions.

Some combination of pre–hazard improvements for hazard recovery is given in Miller-Hooks et al. (2012). The model developed in this paper abstracts a transportation network and allocates resources between preparedness and recovery actions. A two–stage stochastic programming problem is used, and due to integer recourse variables, the integer L–shaped method is required to obtain a solution. The network is subjected to five hazard scenarios, both natural and man–made. Computational issues were detected when the number of nodes and arcs increased beyond certain thresholds.

One of the only exclusively pre–hazard decision models for improving the resilience of the CIS is given by Peeta et al. (2010). A decision–dependent stochastic program is formulated that uses binary decision variables to decide whether to upgrade each link in a road network. The goal of the model is to improve connectivity for first responders. Each upgrade reduces the probability that the link fails, thus increasing the probability that there is a path available for each origin–destination pair. The model is solved using a linear approximation of the objective function and Monte Carlo sampling for approximately one million random instances on a test problem created from data for earthquake plans in Istanbul, Turkey.

### 3.2.2 Stochastic Programming Algorithms

Due to the potentially vast number of scenarios to consider, solving stochastic programming problems often requires the use of specialized solution algorithms. The most common approach used for stochastic programming problems is Benders decomposition, also known as the L–Shaped Method (Birge, 1985; Birge and Louveaux, 1997). The method decomposes the problem into a first stage master problem and a series of subproblems for each possible realization of the uncertainties in the latter stages. The subproblems are used to build an outer linearization of the recourse function for the first stage problem.

A multi–stage stochastic programming problem complicates the use of Benders decomposition because subproblems in intermediate stages are master problems for latter stages. Using a particular
model structure can aid in overcoming this issue. *Block separable recourse* is a strategy that can be exploited for certain specially structured stochastic programming problems that allow multi–stage stochastic programming problems to be treated as two–stage problems (Louveaux, 1986). Block separability exists if the decision variables can be partitioned into two separate sets of decisions, frequently thought of as high–level and detail–level. The detail–level decision are those that do not affect later stages. Then, all high–level decisions can be treated as first stage variables, and detail–level decisions are all treated as second–stage decisions. In this paper, the multi–stage CIS improvement model is developed to leverage this model structure.

Benders decomposition is not generally implemented for speeding convergence to an optimal solution. Instead, its purpose in stochastic programming is to allow consideration of a large number of scenarios which would otherwise not necessarily fit into memory. In fact, naive or simplistic implementations often have slow convergence which has lead to the development of different algorithmic enhancements for faster convergence. Magnanti and Wong (1981) proposed the use of an additional subproblem for each extant subproblem to generate *Pareto optimal cuts*, the strongest cuts possible at each iteration. The cuts are generated by using a *core point*, in the relative interior of the feasible region of the subproblems. A modification of the method was presented by Papadakos (2008) to address the issues that arise from the difficulty of finding multiple core points and numerical unboundedness that can arise in the subproblems. Here, it was proved that approximate core points can sometimes be sufficient to obtain most of the benefits from the Magnanti–Wong enhancements.

### 3.3 CIS Modeling Framework and Multi–Event Extension

In this section a multi–event, multi–stage CIS modeling framework is presented, and the associated resilience metrics that quantify the resilience of the CIS to multiple hazard events and scenarios are presented. Broadly, components of the CIS are modeled as nodes in a network, and relationships between components are represented by links between the nodes. Each component provides services that are used by a population in a community or need by other components to properly function.
The services that components provide are subject to disruption from hazard events. A hazard event consists of a range of potential hazard scenarios; each scenario has a probability of occurring. Resilience metrics quantify the performance of the CIS in each hazard scenario, and the expected performance over all the hazard scenarios is calculated to obtain the performance in that hazard event. In the multi–event framework presented here, the CIS is subjected to multiple hazard events, and its resilience is an aggregated performance over all hazard events.

The modeling framework was originally introduced Piper et al. (2013a) Chapter 2, but example instances of the framework included only a single hazard event. This extension provides instances with multiple hazard events and multiple stages to make improvement decisions. Notation is defined in Table 3.2.

### Table 3.1: Notation for Multi–Event Resilience Metrics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>components of CIS, indexed by $c$</td>
</tr>
<tr>
<td>$e$</td>
<td>events occurred, $e = 0, 1, \ldots, E$</td>
</tr>
<tr>
<td>$H^e$</td>
<td>hazard scenarios possible for event $e$, indexed by $h^e$</td>
</tr>
<tr>
<td>$K$</td>
<td>usage type, indexed by $k$</td>
</tr>
<tr>
<td>$\pi_{h}^e$</td>
<td>probability of hazard scenario $h$ in event $e$</td>
</tr>
<tr>
<td>$TU_c$</td>
<td>set of components whose services can physically reach component $c$</td>
</tr>
<tr>
<td>$TD_c$</td>
<td>set of components physically reachable by services from component $c$</td>
</tr>
<tr>
<td>$SU_c$</td>
<td>set of components with functional linkages going to component $c$</td>
</tr>
<tr>
<td>$SD_c$</td>
<td>set of components with functional linkages starting from component $c$</td>
</tr>
<tr>
<td>$\sigma_{c,k}^e$</td>
<td>base (maximum) service level of usage $k$ provided by component $c$</td>
</tr>
<tr>
<td>$d_{h,c,k}^e$</td>
<td>service type $k$ supplied or demanded at component $c$ in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$f_{h,c,c',k}^e$</td>
<td>potential usage flowing from component $c$ to component $c'$ for service $k$ in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$A_{h}^e$</td>
<td>set of components losing performance directly from hazard in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$B_{h}^e$</td>
<td>set of components losing performance from cascading failures and dependencies in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$\phi_{h,c,k}^e$</td>
<td>initial service level of usage $k$ at component $c$ in scenario $h$ (when $c \in A_{h}$) after event $e$</td>
</tr>
<tr>
<td>$s_{h,c,k}^e$</td>
<td>service level achievable for usage $k$ at $c$ in scenario $h$ after improvements after event $e$</td>
</tr>
</tbody>
</table>

*Notation modified or added from Piper et al. (2013a) (Chapter 2)*
Some preliminary definitions are given here. The set $C$, indexed by $c$, is the components of the CIS. The set $C$ could include road segments, wastewater treatment plants, power lines, emergency facilities like hospitals, etc. The metrics presented assume that the CIS is important for the critical lifeline services it provides. Define the service level of a component as the amount of a particular type of service that a component provides. Let $K$ be the set of types of services, indexed by $k$, (e.g., $k$ could refer to electricity, amount of protection provided by a levee, or the amount of electricity flowing through transmission lines).

Components are connected through directed arcs, called linkages. For components $c_1$ and $c_2$, if there is a linkage $(c_1, c_2)$ that terminates at $c_2$, then $c_1$ is an upstream component of $c_2$; similarly, $c_2$ is downstream of $c_1$. Links are divided into two types: transport links and service links. If a link between components can be used by members of a population to travel, then that link is a transport link. Examples include relationships such as which segments of a road network are connected or the layout of train tracks. Service links connect components if the level of a service type provided by a component influences the service levels of another component, i.e., they are interdependent. These linkages may connect a hospital to the electrical grid, or represent the necessity for a levee to protect components of a road network from flooding.

Sets $TD_c$ and $TU_c$ correspond to components that are physically downstream and upstream respectively from component $c$. Similarly, sets $SD_c$ and $SU_c$ correspond to components that are functionally downstream and upstream respectively from component $c$. The components $c' \in SU_c$ are the components on which $c$ is dependent, and similarly $c' \in SD_c$ are the components that depend on services from $c$.

It is possible, and perhaps necessary, to add a subscript $k, k'$ to each of these four sets and indicate through which service a component has transport or service links to another. For instance, if $c' \in SU_{c,k,k'}$, it would indicate that component $c$ depends on service type $k'$ from component $c'$ to supply its own service type $k$. For example, consider the electricity (usage $k'$) required from a distribution line (component $c'$) that a water treatment plant component (component $c$) requires to supply clean drinking water (usage $k$). For simplicity of notation, these additional indices are usually
3.3.1 Resilience Metrics

Each hazard event consists of multiple potential hazard scenarios that could reduce the functioning of the CIS components. The resilience metrics calculate the expected performance of the CIS in each hazard scenario and aggregate that performance over all the events the CIS is subjected to. Let $\varepsilon$ be the total number of events considered, and $e = 0, \ldots, \varepsilon$ be the number of events that have taken place. Occasionally, the term stage will be used, which is considered to be any of the values $0, \ldots, \varepsilon$, so that stage 0 occurs before event 0, stage 1 occurs after event 1 and before event 2, etc.

Let $s_{h,c,k}^e$ be the service level of service type $k$ available at component $c$ in hazard scenario $h$ after hazard event $e$. This variable is not ever used with $e = 0$, because it is assumed that the CIS is functioning perfectly before any hazard events occur. The set of hazard scenarios possible for event $e$ is given in the set $H^e$. In full, the service at component $c$ is a function of the service of the components in $SU_c$, $s_{h,c,k}^e(SU_c)$, but this dependency is not generally made explicit by notation. In this paper, the service of a component $c$ when affected by the services of its dependencies (as opposed to outright failure) is determined by the component in $SU_c$ with the lowest percentage of its maximum service level, or:

$$s_{h,c,k}^e(SU_c) = \sigma_{c,k} \min_{c' \in SU_c} \left\{ \frac{s_{h,c',k'}^e}{\sigma_{c',k'}} \right\}, \quad \forall c \in C, e = 1, \ldots, \varepsilon, h \in H^e, k, k' \in K \quad (3.1)$$

Note that $k$ and $k'$ may be different service types, and this definition applies only to valid dependencies. As an example of the relationship represented by this equation, if a power plant had only 50% of its normal generating capacity, then a water treatment facility served by that plant could only treat 50% of the water it normally could. If required, different relationships between components could be created.

Resilience metrics for each hazard scenario are calculated based on the service levels of components in the system. The exact mechanism varies for each metric and examples are given later in this
section. Metric values from each hazard scenario are aggregated using an expected value approach that is standard in stochastic programming recourse functions. Other approaches could be used for aggregating scenario performance, such as Value at Risk or expected shortfall. The approach chosen sums the expected value of the metric after each hazard event and then sums over all the events.

As shown in the table, each hazard scenario \( h^e \) will have a probability of occurring, \( \pi_h^e \). Generally, \( \pi_h^e \) would be weighted by the probability of a parent scenario when \( e \geq 2 \) in stochastic programming problems. Instead, a simplified approach is used based on the following observation. When a particular hazard occurs, subsequent natural hazards can be viewed as independent; e.g., the occurrence of a Hurricane Katrina–like event does not imply that the next event will be Hurricane Gustav–like event. Thus, the probability of each hazard scenario does not depend on the probability of a parent, or ancestor, hazard scenario, but is simply the probability of that hazard scenario occurring from all possible hazard scenarios that may occur as that event. This simplification does not weaken the observations gained from applying the framework and considerably lessens the difficulty of optimization models that will be developed in Section 3.3.2 and solved in Section 3.5.

The serviceability and demand satisfaction metrics derived from Piper et al. (2013a) for multi-event resilience metrics are defined as follows. Let \( F \subseteq C \) be a set of components of the CIS whose resilience after a set of potential hazard events is to be measured. Unweighted serviceability for the components of \( F \) is defined as:

\[
s_{CIS} = \frac{1}{\epsilon} \sum_{e=0}^{\epsilon} \sum_{h \in H^e} \frac{1}{|F|} \sum_{c \in F, k \in K} \frac{s_{h,c,k}^e}{\sigma_{c,k}}.
\] (3.2)

Weighted serviceability is given by:

\[
s_{CIS,W} = \frac{1}{\epsilon} \sum_{e=0}^{\epsilon} \sum_{h \in H^e} \frac{1}{|F|} \sum_{c \in C, k \in K} \frac{s_{h,c,k}^e}{\sigma_{c,k}}.
\] (3.3)

Demand satisfaction is used to consider specific types of trips that a population could want to take on a network (for a road network, trip categories might include gas, groceries, or medical supplies). The starting components of such trips are treated as supply components and the destinations
are considered demand components. The metric is calculated for hazard scenario $h$ in event $e$ by solving the multicommodity network flow model presented below:

$$\begin{align*}
\text{minimize} & \quad D(e, h) = \sum_{k \in K} \sum_{c \in C} \left( d_{h,c,k}^e - \sum_{c' \in T_U} f_{h,c',c,k}^e \right) \\
\text{subject to} & \quad \sum_{c' \in T_U} f_{h,c',c,k}^e - \sum_{c' \in T_D} f_{h,c',c,k}^e \geq d_{h,c,k}^e \quad \forall c \in C, k \in K \quad (3.4a) \\
& \quad \sum_{c' \in T_D} f_{h,c',c,k}^e - \sum_{c' \in T_U} f_{h,c',c,k}^e \leq s_{h,c,k}^e \quad \forall c \in C, k \in K \quad (3.4b) \\
& \quad f_{h,c',c,k}^e \geq 0 \quad \forall c \in C, c' \in TD_c, k \in K \quad (3.4c)
\end{align*}$$

It is assumed that there is a dummy “super–supply” component that can be used to satisfy the flow balance constraint, Equation (3.4b). It is also assumed that any flow from super–supply component is not included in the objective function of the model.

If $DS(e, h)$ is the optimal solution obtained by that model for hazard scenario $h$ in event $e$, then the multi–event measurement of the total demand left unsatisfied is given by:

$$\frac{1}{\epsilon} \sum_{e=0}^{\epsilon} \sum_{h \in H^e} \pi_h^e D(e, h). \quad (3.5)$$

Similarly, for the recovery time metric presented in Section 2.2.4, if $R(e, h)$ is the recovery time for hazard scenario $h^e$, the multi–event recovery time is:

$$\frac{1}{\epsilon} \sum_{e=0}^{\epsilon} \sum_{h \in H^e} \pi_h^e R(e, h). \quad (3.6)$$

As the calculation of recovery time implemented depends on the the results of a simulation, the decision models are not extended to optimize for recovery time. The numeric experiments, however, report multi–event recovery times. The implementation of recovery time for this paper uses a dynamic, state–saving, breadth–first graph traversal algorithm to simulate the movement of recovery crews across the CIS network and their effects.
### Table 3.2: Notation for Multi–Stage Optimization Model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U^e_c$</td>
<td>set of possible improvements for component $c$ after event $e$</td>
</tr>
<tr>
<td>$x^e_{c,u}$</td>
<td>binary decision variable, 1 if improvement $u$ is chosen for component $c$ after event $e$ and 0 if not</td>
</tr>
<tr>
<td>$t^e_{h,c,k}$</td>
<td>service level achievable for usage $k$ at $c$ in scenario $h$, limited by service links (when $c \in B_h$) after event $e$</td>
</tr>
<tr>
<td>$\lambda^e_{h,c,u,k}$</td>
<td>benefit in event $e$ for service usage $k$ provided by component $c$ in scenario $h$ if upgrade $u$ is chosen</td>
</tr>
<tr>
<td>$\beta^e$</td>
<td>budget for improvements after event $e$</td>
</tr>
<tr>
<td>$\kappa^e_{c,u}$</td>
<td>cost of improving component $c$ with upgrade $u$ after event $e$</td>
</tr>
</tbody>
</table>

*Notation modified or added from Piper et al. (2013a) (Chapter 2)

### 3.3.2 Decision Model Variables and Dependency Constraints

Improvement decisions for CIS components can increase the service level a component provides in hazard scenarios. Improvements to components could include retrofits of existing components, such as raising the height of a road bed, or design decisions, such as building a water treatment plant in a new location. Improvements are chosen at each stage except the final one, i.e., before and between hazard events and limited by a budget in each stage. The service level provided by a component is a function of service level available from components that it depends on and the improvements selected for the component. Equations to express this relationships use the notation given in Table 3.2.

Improvement decisions are made at each stage, as previously defined. Thus, after event $e$, the binary decision variable $x^e_{c,u}$ is defined for each component $c$ that can be improved, where $x^e_{c,u}$ is 1 when upgrade type $u$ is chosen for component $c$ after event $e$. A decision $x^e_{c,u}$ only effects the events $e, \ldots, \epsilon$. The effect of each decision will vary not only by component, but also by what type of hazard scenario occurs. Thus, the effect of decision variable $x^e_{c,u}$ will be different for hazard scenario in each event that follows. Let $\lambda^e_{h,c,u,k}$ be the effect on the service usage type $k$ for component $c$ in scenario $h$ in event $e$. It is assumed that if a component is improved, then the effect of the improvement in
event $e$ scenario $h$, is the same regardless the in which the improvement was chosen.

Improvements can be implemented for any stage in which they have budget, and the budget after event $e$ has occurred is $\beta^e$. The set of possible upgrades for each component $c$, $U^e_c$, may or may not change after each event, depending on whether or not new technologies for improving CIS components are expected in later budget periods. For the numeric experiments, it was assumed that $U^e_c$ was identical for all $e$. The cost of improvement $u$ for component $c$ before event $e$ is given by $\kappa^e_{c,u}$. For the numeric examples it was assumed that technology did not change and thus the cost of each improvement was constant for all stages.

Discrete investments made for a component $c$ will increase the amount of service available from a component in hazard scenarios when the component would otherwise provide less service. Let $x^e_{c,u}$ be a binary variable, equal to 1 when component $c$ receives improvement $u$ before $e + 1$ events have occurred. Furthermore, it is assumed that each component receives at most a single upgrade, written as:

$$
\sum_{e=0}^{\epsilon-1} \sum_{u \in U^e_c, u \neq 0} x^e_{c,u} = 1 \quad \forall c \in C.
$$

(3.7)

It is also assumed that level 0 is the “do–nothing” decision that is available for all components, costs nothing, and has no effect.

Dependency constraints connecting service at component $c$ to the service of the components in $SU_c$, as shown in Equation (3.1), are accommodated using the auxiliary variable $t^e_{h,c,k}$. This variable represents the highest service level component $c$ can obtain in event $e$ and hazard scenario $h \in H^e$ based on the service levels of the components in $SU_c$. In other words, $t^e_{h,c,k}$ is the highest service level that $c$ can obtain in that hazard scenario without any improvements. Thus, the multi–event optimization framework has the constraint:

$$
t^e_{h,c,k} \leq \sigma^{e}_{c,k} \left( \frac{s^e_{h,c,k'}}{\sigma^{e}_{c',k'}} \right), \quad \forall c \in C, \forall c' \in SU_c, \forall e = 1, \ldots, \epsilon, h \in H^e, k', k \in K.
$$

(3.8)

If $\lambda^e_{h,c,u,k}$ is the additional service level for usage type $k$ added to component $c$ in hazard scenario
\( h \in H^e \) for upgrade \( u \), then the service level of component \( c \), after adjusting for any improvement chosen for \( c \), is:

\[
s_{h,c,k}^e \leq t_{h,c,k}^e + \sum_{e'=0}^{e-1} \sum_{u \in U^e_c} x_{c,u}^{e',h,c,u,k} \quad \forall c \in C, e = 1, \ldots, e, h \in H^e, k \in K
\] (3.9)

It is assumed that improvements cannot add to the base service level of a component, but instead can increase the minimum service level that a component can provide if it fails, thus, it is necessary to add a constraint that \( s_{h,c,k}^e \leq \sigma_{c,k} \) for each hazard scenario. This constraint could be relaxed if desired.

To represent the effects of hazard scenarios, for failing components \( (c \in A_{h,k}^e) \), \( t_{h,c,k}^e \) is constrained so that:

\[
t_{h,c,k}^e \leq \phi_{h,c,k}^e
\] (3.10)

where \( 0 \leq \phi_{h,c,k}^e \leq \sigma_{c,k} \) a random variable for the service level of component \( c \) for usage type \( k \) in the hazard in scenario \( h \). For example, a collapsed bridge would have \( \phi_{h,c,k}^e = 0 \), and a power plant operating at 50% capacity would have \( \phi_{h,c,k}^e = 0.5 \).

### 3.3.3 Optimization Model Instance

In this section, a multi–event optimization model is presented to choose improvements over multiple stages that increase the amount of demand satisfied for particular types trips on a road network. The optimization model chooses what type of improvement decisions each CIS component receives (if any) in a given stage subject to the improvement budget in that stage. After improvements are selected the CIS is subjected to a hazard event that has a range of possible hazard scenarios, with a probability distribution determining the likelihood of each hazard scenario. If a component loses service level in a particular hazard scenario due to interdependencies or directly from the storm, then an improvement decision made for the component potentially increases the service level that the component provides in that scenario as high as its base service level.
After improvement effects are determined for each component, constraints ensure that the effects of interdependencies on service levels are respected in each hazard scenario. After the expected demand satisfied is calculated for hazard event, the process repeats, and improvement decisions for the next stage are selected with a new budget to find additional enhancements to protect the CIS. The improved CIS is subjected to another hazard event with a range of hazard scenarios, until the desired number of hazard events occurs. The objective function of the optimization model averages the expected demand satisfied in each event to produce the demand satisfaction resilience metric.

An example instance of the multi-stage stochastic programming problem for the demand satisfaction metric is presented here, using the notation from Tables 3.1 and 3.2. This model represents a CIS consisting of a levee and road system, where the levee protects certain road components from flooding, as in Figure 3.1. The components of the road system are segments of the road network and represented by the set $R$. Segments of the levee represent the components of the levee system and these components are represented by the set $L$. Service usage types, $K$, are divided into two categories, one for service usage between components that are endpoints of service links, $K_S$, and a category for usage types indicating how the populace uses the infrastructure, given by $K_T$. 
Road components provide service level as capacity in vehicles/hour, which might be anywhere from a few hundred to several thousand if there is no degradation. Road capacity will be service type $k_0 \in K_S$. Road components have transport links to adjacent road components (sets $TD_c$ and $TU_c$). Road components also have service links to the levee components that protect them, so that for all $c \in R$, $SU_c \subset L$. Similarly, if $c$ is a levee component, then $SD_c \subset R$.

A component of the levee, $c$, has service level equal to the amount of protection it provides to the road components, which is expressed as a percentage. The levee service type is indicated as $k_1 \in K_S$. A lower service level indicates that the levee has been breached or overtopped, which causes flooding and reduces the road capacity. For instance, a loss of 10% of levee service at component $c$ causes a loss of at least 10% of vehicles/hour for the road components $c' \in SD_c$.

The second category of service usage, set $K_T$, consists of the types of trips a populace wants to make on the road system, e.g., consumer goods shopping, commuting, or product shipment. They can be considered equivalent to the commodities in a multi–commodity flow problem over the transport links of the road network. Thus, for a component of the road network,

$$\sum_{c' \in TD_c} \sum_{k \in K_T} f_{h,c',c,k} \leq s_{h,c,k_0}^e,$$  \hspace{1cm} (3.11)

which matches Equation (3.4c). Supply and demand for a particular usage, $d_{h,c,k}^e$, correspond to the number of vehicles from $c \in R$ that want to undertake a trip of type $k \in K_T$, or the number of vehicles that can have type $k$ trip completed at component $c$. It is assumed that vehicles start at components with supply and transport to components that have demand, though the opposite could be logical.

Finally, to ensure that there is always a feasible transport flow in every hazard scenario, it is assumed that there is a dummy "super–supply" component, $c_{Sup}$, in the road network that can fulfill any amount of demand. Variable $f_{h,c_{Sup},c,k}^e$ is the amount of demand unsatisfied at component $c$ of usage type $k$. Flows from the super–supply component are added as needed to the flow constraints.
A full model formulation for this instance follows.

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{\epsilon} \sum_{e=1}^{\epsilon} \sum_{h \in H^e} \pi_h^e \sum_{k \in K_T} \left( \sum_{d_{h,c,k}^e \geq 0} d_{h,c,k}^e - \sum_{c' \in T U_c} f_{h,c',c,k}^e \right) \\
\text{subject to} & \quad \sum_{c' \in T U_c} f_{h,c',c,k}^e - \sum_{c' \in T D_c} f_{h,c,c',k}^e \geq d_{h,c,k}^e \quad \forall c \in R, k \in K_T, e \in \{1, \ldots, \epsilon\}, h \in H^e \\
& \quad \sum_{k \in K_T} \sum_{c' \in T D_c} f_{h,c,c',k}^e \leq s_{h,c,k}^e \quad \forall c \in R, e \in \{1, \ldots, \epsilon\}, h \in H^e \\
& \quad t_{h,c,k}^e \leq \phi_{h,c,k}^e \quad \forall e \in \{1, \ldots, \epsilon\}, h \in H^e, c \in A_h^e, i = 0,1 \\
& \quad t_{h,c,k}^e + \sum_{e' = 0}^{e-1} \sum_{u \in U_c^e} x_{c,u}^e \lambda_{h,c,u,k}^e \geq s_{h,c,k}^e \quad \forall e \in \{1, \ldots, \epsilon\}, h \in H^e, c \in C, i = 0,1 \\
& \quad t_{h,c,k}^e \leq \sigma_{c,k}^e \left( \frac{s_{h,c,k}^e}{\sigma_{c',k}^e} \right) \quad \forall c \in R, c' \in SU_c, e = 1, \ldots, \epsilon, h \in H^e \\
& \quad s_{h,c,k}^e \leq \sigma_{c,k}^e \quad \forall c \in C, e \in \{1, \ldots, \epsilon\}, h \in H^e, i = 0,1 \\
& \quad \sum_{c \in C} \sum_{u \in U_c^e} \kappa_{c,u}^e x_{c,u}^e \leq \beta^e \quad \forall e \in \{0, \ldots, \epsilon - 1\} \\
& \quad \sum_{e=0}^\epsilon \sum_{u \in U_c^e, u \neq 0} x_{c,u}^e = 1 \quad \forall c \in C \\
& \quad f_{h,c',c,k}^e \geq 0 \quad \forall c \in R, c' \in T D_c, k \in K_T \\
& \quad s_{h,c,k}^e, t_{h,c,k}^e \geq 0 \quad \forall e = 1, \ldots, \epsilon, h \in H^e, c \in C, i = 0,1 \\
& \quad x_{c,u}^e \in \{0,1\} \quad \forall c \in C, u \in U_c^e, e \in \{0, \ldots, \epsilon - 1\}
\end{align*}
\]

The objective function, Equation (3.12a), calculates the total expected unsatisfied demand after all events for each of the trip types on the road network. Equivalently, the objective function could make use of the “super-supply” nodes and variables and simplify to:

\[
\text{minimize} \quad \frac{1}{\epsilon} \sum_{e=1}^{\epsilon} \sum_{h \in H^e} \pi_h^e \sum_{k \in K_T} \sum_{d_{h,c,k}^e \geq 0} f_{h,c,\text{sup},c,k}^e \quad (3.13)
\]
a minimum level of serviceability or as the objective function.

The constraints (3.12b) and (3.12c) ensure that the necessary vehicle travel demand is met and that the number of vehicles leaving a road component is limited by the component capacity in that hazard scenario and event. These network flow and capacity constraints would be omitted if a different metric was chosen.

Constraints (3.12d)–(3.12f) are used to set the capacity of the road system and functionality of the levee components for each hazard scenario. Constraint (3.12d), like Equation (3.10), constrains the unimproved service of any component that fails to the service level of the failure. Failure service level $\phi_{h,c,k}$ for component $c$ will be between zero and $\sigma_{c,k}$ and applies to both road and levee segments, as both may fail directly from a natural hazard. Constraint (3.12e) ensures that the actual capacity of a road segment is limited by the unimproved service plus the effects of any improvement that was chosen for that component. The constraint (3.12f) ensures that the capacity of a road component is limited by the functionality of levee components that protect it.

The budget for improvements in each stage is given in Constraint (3.12h). Constraint (3.12i) enforces the assumption that each component can receive at most one upgrade during all stages. Constraints (3.12j), (3.12k), and (3.12l) are nonnegativity and binary decision variable constraints for flow variables, service level variables and improvement decisions.

The model was verified on a small example with four road components, three levee components, and two events with two hazard scenarios each (including the “nothing happens” hazard scenario). The small example also had two types of trips with each represented by a single trip on the network. Initially, the model was checked with one hazard event, and the second one was added after model behavior appeared to be correct. Improvement plans with different combinations of improvements were generated. Component service levels in each improvement plan were compared and found to agree with the expected results. Additionally, the demand unsatisfied obtained by the model was verified for each hazard scenario under each improvement plan.
3.4 Solution Approach

Depending on the size of the CIS and the number of hazard scenarios and stages, the model presented in the preceding section may be solved by standard techniques in commercial solvers. In Section 3.5, certain instances will be solved in this way. For the more difficult instances, a Benders decomposition method, commonly called the L-shaped method in stochastic programming, will be used to solve multi-event instances for the demand satisfaction metric. While the Benders decomposition generally requires modification for a multi-stage stochastic programming problem, the particular structure of the model shown in Equations (3.12a)–(3.12l) can be exploited to solve the problem as if it was a two-stage problem.

If all improvement decisions, $x_{e,c,u}$, are viewed as high-level, planning and strategy decisions, and the remaining variables, used only to calculate metrics, are viewed as detailed, operational decisions, then the model can be reformulated with block-separable recourse (Louveaux, 1986). In essence, modeling the problem with block separability views the engineering improvements chosen as being entirely a priori decisions, regardless of the operations or status of the CIS in each event and scenario. Using block-separable recourse also requires using the multicut version of Benders decomposition for stochastic programs. Another benefit of exploiting block-separable recourse is that all integer variables will be in the first-stage, which eliminates the need for more complex algorithms.

The subproblem formulation for determining demand satisfaction in a particular hazard scenario is presented below. This subproblem is for the levee and road CIS given earlier and for a particular pair $(e, h)$ of event and scenario. The improvement decisions are written as $\hat{x}^e_{c,u}$ to indicate that
when solving a subproblem, improvement variables are fixed (externally, by a master problem).

\[
\begin{align*}
\text{minimize} & \quad \sum_{k \in K_T} \left( \sum_{c \in R} \sum_{c' \in T_U} d_{h,c,k} - \sum_{c' \in T_U} f_{h,c',k} \right) \\
\text{subject to} & \quad \sum_{c' \in T_U} f_{h,c',k} - \sum_{c' \in T_D} f_{h,c',k} \geq d_{h,c,k} \quad \forall c \in R, k \in K_T \\
& \quad \sum_{k \in K_T} \sum_{c' \in T_D} f_{h,c',k} \leq s_{h,c,k_0} \quad \forall c \in R \\
& \quad t_{h,c,k_i}^e \leq \phi_{h,c,k_i}^e \quad \forall c \in A^e_h, i = 0, 1 \\
& \quad t_{h,c,k_i}^e + \sum_{e'=1}^{e-1} \sum_{u \in U_f} x_{h,c,u,k_i}^{e'} \geq s_{h,c,k_i}^e \quad \forall c \in B^e_h, i = 0, 1 \\
& \quad t_{h,c,k_0}^e \leq \sigma_{c,k_0} \left( \frac{s_{h,c,k_i}^e}{\sigma_{c,k_i}} \right) \quad \forall c \in R, \forall c' \in SU_e, \forall e = 1, \ldots, e, h \in H^e \\
& \quad s_{h,c,k_i}^e \leq \sigma_{c,k_i} \quad \forall c \in C, e \in \{1, \ldots, e\}, h \in H^e, i = 0, 1 \\
& \quad f_{h,c',k}^e \geq 0 \quad \forall c \in R, c' \in TD_e, k \in K_T
\end{align*}
\]

(3.14a) (3.14b) (3.14c) (3.14d) (3.14e) (3.14f) (3.14g) (3.14h)

To determine the feasibility and optimality cuts for the Benders decomposition, it is helpful to examine the constraint matrix for the subproblems. The constraint matrix with the corresponding constraints for each section, ignoring slack variables, is given in Figure 3.2.

<table>
<thead>
<tr>
<th>Constraint RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{h,c,k}$</td>
</tr>
<tr>
<td>$0$</td>
</tr>
<tr>
<td>$s_{h,c,k}^e$</td>
</tr>
<tr>
<td>$t_{h,c,k}^e$</td>
</tr>
<tr>
<td>$x_{h,c,u}^e$</td>
</tr>
<tr>
<td>$\phi_{h,c,k}^e$</td>
</tr>
<tr>
<td>$\sigma_{h,c}^e$</td>
</tr>
</tbody>
</table>

Figure 3.2: Constraint Matrix for Demand Satisfaction Optimization Model
$F_C$ is the constraint matrix for conservation of flow and $F_S$ is the constraint matrix for flow limited by service levels. $I_{|C|}$ corresponds to a $|C| \times |C|$ identity matrix. $\Lambda^e_h$ is the matrix of decision effects for CIS components in scenario $h$ after hazard event $e$. $M$ and $N$ are both $|\mathcal{C}| \cdot \sum_{c \in \mathcal{C}} |SU_c| \times |\mathcal{C}|$ matrices. The $c^{th}$ column of each corresponds to variables $s^e_{h,c,k}$ and $t^e_{h,c,k}$, respectively. When entry $i, c$ in $N$ has coefficient 1, entry $i, c'$ in $M$ has value $-\frac{\sigma_{c,k_1}}{\sigma_{c',k_1}}$, where $c' \in SU_c$.

Let $p_i$ be the vector of the dual solutions for constraint set $i$, where $i$ is given by the equation numbers (e.g., $p^e_h$, (3.14g)) in scenario $h$, stage $e$.

In the standard Benders decomposition, cuts added to the master problem are derived using $\lambda^e_h$ to determine and an estimate of the dual subproblems’ optimal values to determine the right-hand side of the cut. The structure of the subproblem's constraints, however, allows for some efficiencies. For a given stage $e$ and scenario $h$, the coefficient for cuts generated in iteration $i$ assuming $\hat{x}^i$ is the given master problem solution, $\Omega_{e,h}(\hat{x}^i)$, is defined as

$$\Omega_{e,h}(\hat{x}^i) = \pi^e_h p^e_{h,(3.14e)} \Lambda^e_h.$$ (3.15)

The corresponding right–hand side of cuts, $\omega_{e,h}(\hat{x}^i)$, is given as:

$$\omega_{e,h}(\hat{x}^i) = \pi^e_h \left( p^e_{h,(3.14b)} d^e_{h,c,k} + p^e_{h,(3.14d)} \phi^e_{h,c,k} + p^e_{h,(3.14g)} \sigma^e_{h,c} \right).$$ (3.16)

Thus, the cut added from subproblem $(e, h)$ in iteration $i$ is $\Omega_{e,h}(\hat{x}^i) X + \theta^e_h \leq \omega_{e,h}(\hat{x}^i)$, where $X$ is a vector of all possible improvement decisions, and $\theta^e_h$ is the variable that approximates the resilience of the event $e$ hazard scenario $h$. The master problem at iteration $n$ is:

$$\text{minimize} \quad \frac{1}{\varepsilon} \sum_{e=1}^{\varepsilon} \sum_{h \in H} \theta^e_h$$ (3.17a)

$$\text{subject to} \quad \sum_{c \in \mathcal{C}} \sum_{u \in U^e_c} \kappa^e_{c,u} x^e_{c,u} \leq \beta^e_e \quad \forall e \in \{0, \ldots, \varepsilon - 1\}$$ (3.17b)

$$\sum_{e=0}^{\varepsilon} \sum_{u \in U^e_c, u \neq 0} x^e_{c,u} = 1 \quad \forall c \in \mathcal{C}$$ (3.17c)
\[ \Omega_{e,h}(\hat{x}^i) X + \theta_{e,h} \leq \omega_{e,h}(\hat{x}^i) \quad \forall i = 1, \ldots, n - 1 \tag{3.17d} \]
\[ x_{c,u}^e \in \{0, 1\} \quad \forall c \in C, u \in U_c^e, e \in \{0, \ldots, \epsilon - 1\} \tag{3.17e} \]

Equation (3.17d) are the optimality cuts from solving the Benders decomposition subproblem. Feasibility cuts are unnecessary, since the super-supply node guarantees complete recourse for the demand satisfaction metric (for the serviceability metric, complete recourse is trivially guaranteed).

While it would be preferable to solve the extensive form problem if possible, the size of the problem and desire to consider many hazard scenarios necessitates Benders decomposition for certain model instances. Solving this problem through Benders decomposition still has issues. Already, there is the issue of the number of binary variables in the master problem, which is \(|E| \sum_{c \in C} |U_c|\), the number of stages times the number of possible improvements for each component. Decomposing the problem removes constraint information from the master problem that could be used to generate efficient cuts for the discrete variables. It will be shown that this is the case for the serviceability problem. Another issue is frequent slow convergence of Benders decomposition. An enhancement that was added to the algorithm is described below.

### 3.4.1 Special Cuts for Benders Decomposition

The second issue is the well-known slow convergence of Benders decomposition (Bertsimas and Tsitsiklis, 1997). Additional improvements to the algorithm can be made through the use of special cuts, first given by Magnanti and Wong (1981). First, necessary notation is introduced. This notation is separate from that used to develop the CIS optimization models. The notation is adapted from Papadakos (2008), which contains a more detailed explanation of the material presented here.
Given the following problem:

$$\begin{align*}
\max & \quad c^T y + f(x) \\
\text{s.t.} & \quad Ay + x = b \\
& \quad y \geq 0 \\
& \quad x \in X
\end{align*}$$  \tag{3.18a}

where it is assumed that $X \subset \mathbb{R}^m$ is the set of possible combinations of the discrete variables $x$, $A \in \mathbb{R}^{m \times n}$, $y$ and $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $f(\cdot)$ is the cost function for the discrete variables. Then, given discrete variables $x$ that are fixed at $\hat{x}$, Benders decomposition creates a subproblem of the form:

$$\begin{align*}
\max & \quad c^T y \\
\text{s.t.} & \quad Ay = b - \hat{x} \\
& \quad y \geq 0
\end{align*}$$  \tag{3.19a}

with the dual problem:

$$\begin{align*}
\min & \quad (b - \hat{x})p \\
\text{s.t.} & \quad A^T p \leq c
\end{align*}$$  \tag{3.20a}

where $p \in \mathbb{R}^m$ are the dual variables.

The master problem is:

$$\begin{align*}
\max & \quad f(x) + \theta \\
\text{s.t.} & \quad (b - x)p(\hat{x}) \geq \theta \quad \forall p(\hat{x}) \in P \\
& \quad x \in X
\end{align*}$$  \tag{3.21a}

where $p(\hat{x})$ is the optimal dual solution obtained from the subproblem defined by $\hat{x}$ and $P$ is the set
of dual variables for the subproblems that have been solved so far. Equation (3.21b) is the optimality cuts that have been added to the problem. Feasibility cuts are not addressed here, but their form is similar.

Using this notation, a cut is defined to be a *nondominated* cut, and therefore the strongest cut possible, if the following holds.

**Definition 1.** A cut given by a particular solution to the dual problem, \( p_1(\hat{x}) \in P \), dominates another \( p_2(\hat{x}) \in P \), if

\[
(b - x)^T p_1(\hat{x}) \leq (b - x)^T p_2(\hat{x}), \quad \forall x \in X,
\]

and there exists one point \( x \in X \) for which the inequality is strict. (Magnanti and Wong, 1981)

Note that it only makes sense to search for nondominated cuts if subproblems have multiple solutions.

Assuming that \( z(\hat{x}) \) is the optimal objective function value for the subproblem given in Equations (3.20a)–(3.20b), then Magnanti and Wong (1981) showed that nondominated cuts can be obtained by solving the problem:

\[
\begin{align*}
\min & \quad (b - x_0)^T p \\
\text{s.t.} & \quad A^T p \leq c \\
& \quad (b - x)^T p = z(\hat{x}),
\end{align*}
\]

where \( x_0 \) is a *core point*, one in the relative interior of the convex hull of \( X \). The cuts obtained by this method are called *Magnanti–Wong (MW)* cuts.

The method is complicated by the fact that finding a core point can be difficult. Papadakos (2008) enhanced the Magnanti–Wong method by proving that Equation (3.25) is unnecessary for generating a nondominated cut provided that \( x_0 \) is a core point of \( X \). Even in the case where \( x_0 \) is not a core point, it was shown that using approximate core points with MW cuts can improve over a standard Benders
decomposition implementation.

Given a particular solution $x^i$ for iteration $i$, Papadakos (2008) approximated the core point, $x_0$, using the following formula to update the core point at every iteration:

$$x_0 = \frac{1}{2} x_0 + \frac{1}{2} \hat{x}^i$$  \hspace{1cm} (3.26)

where $\hat{x}^i$ is the solution to the master problem at iteration $i$. With this update, even if the initial $x_0$ is not a core point, the sequence converges to a core point.

Even in the case where $x_0$ is not a core point, it is possible to use alternative points that would generate nondominated cuts if the requirements of the following Corollary are satisfied.

**Corollary** (9). If $\text{cone}(b - X) \supseteq X$, $c \geq 0$, and there is a $y \geq 0$ such that $Ay = b$, then for every $x \in X$ and $0 \leq \lambda < 1$, $\lambda x$ is a point that generates a nondominated cut. (Papadakos, 2008)

To use these special cuts without a need to generate core points, it is necessary for the optimization problem to fulfill the requirements of Papadakos’ Corollary. It is now shown that the CIS optimization framework satisfies the requirements.

**Satisfaction of Papadakos’ Corollary.** Each requirement is individually shown to be true for the subproblems in Equations (3.14).

1. $c \geq 0$. Every constant term can be dropped in Equation (3.14a) for the demand satisfaction metric, and the minimization can be converted to a maximization by multiplying the objective function by negative one. Every resulting coefficient will be greater than or equal to 0. For the serviceability metric, this is already trivially true. Another possibility for the demand satisfaction optimization model is to use the alternate objective function given in Equation (3.13).

2. $\text{cone}(b - X) \supseteq X$. Let $X$ be all possible combinations of improvement effects for a subproblem (so that an element $x \in X$ is a subset of the entries of $\Lambda$). Also, let $b$ be the right–hand side of the subproblem. Most dimensions of $\text{cone}(b - X)$ simplify to $\text{cone}(b - 0)$, whenever the effects
of improvements for a constraint are always zero. For these dimensions, it is trivially true that $\text{cone}(b - 0) \supseteq \{0\}$. The simplification applies to each of Equations (3.14b)–(3.14h) except for Equation (3.14e).

Some additional work is required to show that this will hold for the constraints given by Equation (3.14e). Rewritten in standard form, the constraints are:

$$s^e_{h,c,k_i} - t^e_{h,c,k_i} - \sum_{e'=0}^{e-1} \sum_{u \in U_{E}^e} x^e_{c,u} \lambda^e_{h,c,u,k_i} \leq 0. \quad (3.27)$$

Each of the $\lambda^e_{h,c,u,k_i}$ effects are positive, so their coefficients in this standard form are negative. Thus, each $x \in X$ will have a negative value for these constraints. The cone formed by $b - X$ is

$$\text{cone}(b - x) = \text{cone}(0 - (-\lambda^e_{h,c,u,k_i}))$$
$$= \text{cone}(\lambda^e_{h,c,u,k_i})$$
$$= [0, \infty) \quad (3.30)$$

which obviously is not a superset of $X \subset \mathbb{R}^{\leq 0}$. As suggested by Papadakos (2008), the values of $X$, and therefore the coefficients of $\Lambda$, will be translated so that $\text{cone}(b - X) \supseteq X$.

Let $M = \max_{h,c,k,u} \{\lambda^e_{h,c,u,k_i}\}$. A constraint equivalent to Equation (3.14e) will be used instead:

$$s^e_{h,c,k_i} - t^e_{h,c,k_i} - \sum_{e'=0}^{e-1} \sum_{u \in U_{E}^e} x^e_{c,u} (\lambda^e_{h,c,u,k_i} - M) \leq M. \quad (3.31)$$

It is valid to offset the $-M$ added to the service effect coefficient for each upgrade possibility with a single $M$ on the right-hand side, because only one improvement can be chosen for each component, so at most the necessary adjustment is $M$. After this adjustment, $b = M$ for each component, and the coefficients on each decision variable are $-\lambda^e_{h,c,u,k_i} + M \in [0, M]$. The result is that the adjusted $X$ becomes a subset of nonnegative values. Considering the cone of
$b - X$ now gives:

$$cone(b - x) = cone(M - (-\lambda_{h,c,u,k_i} + M)$$

$$= cone(\lambda_{h,c,u,k_i})$$

$$= [0, \infty).$$

As $[0, \infty] \supset X$, this requirement for the corollary is satisfied.

3. The final requirement is that there exists a $y$ so that $Ay = b$. This is trivially true for the demand satisfaction or serviceability subproblems, as it is equivalent to requiring that the subproblem is feasible when no improvements are chosen, which is always the case.

$$\Box$$

As a result, MW cuts were implemented and compared against a Benders decomposition without the use of these cuts. The necessary adjustment to Constraint (3.14e) was implemented.

Other enhancements to Benders decomposition were explored, such as first solving with relaxed integer variables until convergence and including the addition of various cuts to try to reduce the number of times the master problem needed to be solved. The relaxed problem was solvable, but when the relaxation was removed, the resulting integer programming problem proved intractable due to running out of memory during the branch and cut process used to solve the master problem.

### 3.5 Numeric Experiments

The numeric experiments are conducted on an illustrative example based on a real infrastructure system from a coastal city community, whose fictional adaptation is named Coastal City, and is shown in Figure 3.1. This version of Coastal City is slightly modified from the original presented in Section 2.3, as some parts of the road network and some levee components have been removed. This city faces annual possibilities of natural hazards in the form of hurricanes. In Coastal City, a network of levees protect a road network from storm–surge–caused flooding. The CIS of Coastal City
has 1,122 road components in the network and 64 components in the levee that protect them. The model presented for levees and roads (Equations (3.12)) will be used for the demand satisfaction optimization. A similar model is used for serviceability optimization. Additionally, the road network has 1,448 trips on it, which are divided into 15 categories.

Two problem instances will be considered. Both are two–event (two–stage) instances. The instance \( A \) has four first–event hazard scenarios and four second–event hazard hazards, both events include a “no–hazard scenario” that occurs 20% of the time. Problem instance \( B \) has thirty-one hazard scenarios in the first and second events, again with a “no–hazard scenario” that occurs 20% of the time in each event. For both problem instances demand at a component, \( d^e_{h,c,k} \), is identical for all \( e \) and \( h \). Additionally, \( \phi^e_{h,c,k} \) is 0.0 for each failing component, rather than some intermediate level between 0.0 and \( \sigma_{c,k} \).

Each component has three possible levels of improvement decisions, including the “do–nothing” decision. Improvement effects and costs were randomly generated for both instances, and budget was set to three million dollars in each instance (though this may not be reflective of the resources available in the actual CIS). For both instances \( A \) and \( B \), this creates 7,116 binary variables. There were some improvements for which no \( \lambda^e_{h,c,u,k} \) were nonzero. These decisions were fixed at 0, reducing the number of binary variables to 5,362 in problem instance \( A \) and 6,580 for instance \( B \).

After solutions are obtained for each of the models, all metrics are calculated for the CIS based on the improvements that were chosen. This includes the serviceability, demand satisfaction, serviceability recovery time, and demand satisfaction recovery time, though optimization models are only run for the serviceability and demand satisfaction metrics. The Benders decomposition algorithm is used only for the demand satisfaction optimization on problem instance \( B \). A time limit of 24 hours, with a maximum of one hour spent solving each master problem solution, was set for this approach.

Two measurements of the utility of the stochastic programming approach are reported: the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS). Both are defined in Birge and Louveaux (1997), and explanations are included here. The notion of the recourse problem (RP) solution is necessary for both measurements. The RP solution is the objective function.
value of the optimal solution of a stochastic program.

To define EVPI, the concept of the \textit{Wait–and–See} solution (WS) is needed. The WS solution is the optimal solution for a stochastic program where a separate set of first–stage decisions and constraints exist for every scenario that occurs. Thus, each scenario has its own set of optimal decisions and an optimal performance for that scenario. If $z_h^e$ is the optimal performance in hazard scenario $h$ and event $e$ given by the set of decisions made for that scenario, then $1/\varepsilon \sum_{e=1}^{E} \sum_{h \in H} \pi_h^e z_h^e$ is the WS solution value. The name comes from the fact that the model can choose first–stage decisions specific to each scenario that might occur, as if it is possible to wait and decide which first–stage decisions to select after it is known what will happen. EVPI is defined the difference between the WS solution and the RP solution. It measures how far the RP is from the best performance that could be obtained if the exact scenario to occur were known. Metrics beyond the objective function value for the WS solution are not reported, as the WS solution is a set of solutions rather than a single solution.

The value of the stochastic solution requires defining the \textit{expected value} (EV) solution, which is created by replacing the hazard scenarios in each hazard event with a single hazard scenario. The single hazard scenario sets the maximum service level due to failures for each component, $\phi_{h,c,k}$, to its expected value, $\sum_{h \in H} \pi_h \phi_{h,c,k}$. The EV solution is obtained by solving this expected value problem. EEV is defined as the expected performance from using the first–stage decisions of the EV solution in the RP. In the Coastal City resilience problem, EEV is obtained by fixing all improvement decisions based on the EV solution. VSS is defined as the difference between the RP solution and EEV solution. VSS measures the solution improvement gained by considering the different scenarios that may occur.

The runs performed for each problem instance and their solution approach(es) are summarized in Table 3.3. A “Standard” solution approach indicates that the extensive form of the problem was used to solve it, rather than Benders decomposition.

Two versions of Benders decomposition were run with Magnanti–Wong cuts. The first version, MW, uses an initial approximate core point obtained by setting all decisions to the middle improvement level. The second, MW+, attempts to use Corollary 9 from Papadakos (2008) and uses
Table 3.3: Numeric Experiment Settings

<table>
<thead>
<tr>
<th>Problem Instance</th>
<th>Metric</th>
<th>Optimization Model Instance</th>
<th>Solution Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Serviceability</td>
<td>EEV</td>
<td>Standard</td>
</tr>
<tr>
<td>A</td>
<td>Serviceability</td>
<td>RP</td>
<td>Standard</td>
</tr>
<tr>
<td>A</td>
<td>Serviceability</td>
<td>WS</td>
<td>Standard</td>
</tr>
<tr>
<td>A</td>
<td>Demand Satisfaction</td>
<td>EEV</td>
<td>Standard</td>
</tr>
<tr>
<td>A</td>
<td>Demand Satisfaction</td>
<td>RP</td>
<td>Standard</td>
</tr>
<tr>
<td>A</td>
<td>Demand Satisfaction</td>
<td>WS</td>
<td>Standard</td>
</tr>
<tr>
<td>B</td>
<td>Serviceability</td>
<td>EEV</td>
<td>Standard</td>
</tr>
<tr>
<td>B</td>
<td>Serviceability</td>
<td>RP</td>
<td>Standard</td>
</tr>
<tr>
<td>B</td>
<td>Serviceability</td>
<td>WS</td>
<td>Standard</td>
</tr>
<tr>
<td>B</td>
<td>Demand Satisfaction</td>
<td>EEV</td>
<td>Standard</td>
</tr>
<tr>
<td>B</td>
<td>Demand Satisfaction</td>
<td>RP</td>
<td>Benders decomposition</td>
</tr>
<tr>
<td>B</td>
<td>Demand Satisfaction</td>
<td>RP</td>
<td>Benders with MW Cuts (MW)</td>
</tr>
<tr>
<td>B</td>
<td>Demand Satisfaction</td>
<td>RP</td>
<td>Benders with MW Cuts and Corollary 9 (MW+)</td>
</tr>
<tr>
<td>B</td>
<td>Demand Satisfaction</td>
<td>WS</td>
<td>Standard</td>
</tr>
</tbody>
</table>

72
Table 3.4: Metrics Results for Problem Instance A

<table>
<thead>
<tr>
<th>Model</th>
<th>Metrics After Improvements</th>
<th>Serviceability Demand Satisfaction</th>
<th>Serviceability Recovery Time</th>
<th>Demand Satisfaction Recovery Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do Nothing</td>
<td></td>
<td>64.36%</td>
<td>42.44%</td>
<td>39.47</td>
</tr>
<tr>
<td>EEV (Service)</td>
<td></td>
<td>*71.28%</td>
<td>63.26%</td>
<td>39.62</td>
</tr>
<tr>
<td>EEV (Demand)</td>
<td></td>
<td>71.00%</td>
<td>*61.54%</td>
<td>39.62</td>
</tr>
<tr>
<td>RP (Service)</td>
<td></td>
<td>*71.62%</td>
<td>71.09%</td>
<td>34.86</td>
</tr>
<tr>
<td>RP (Demand)</td>
<td></td>
<td>75.38%</td>
<td>*73.84%</td>
<td>38.95</td>
</tr>
<tr>
<td>WS (Service)</td>
<td></td>
<td>*80.95%</td>
<td>*100%</td>
<td></td>
</tr>
</tbody>
</table>

an initial core point obtained by setting all decisions to the highest improvement level available and translates the effects of improvements according to Equation (3.31). Both use Equation (3.26) to update the approximate core points.

For each optimization model, solution times are also reported. Solution times are shown individually for master problems, subproblems, and Magnanti–Wong subproblems when applicable.

Models and the Benders decomposition algorithm were implemented in C++ with CPLEX 12.4. Most problems were run on a machine with eight cores running at 2.8GHz and 8GB of RAM on Ubuntu 12.04. For the demand satisfaction problems, both the extensive form and Benders decomposition, memory was a concern, and those problems were run on a single node of a cluster that had thirty–two cores running at 800 Mhz and 64GB of RAM.

3.5.1 Results

Table 3.4 shows the resilience metrics of Coastal City problem instance A. Each row displays the resilience metrics obtained in Coastal City under different improvement parameters and goals. For instance, EEV (Service) shows the metric values when the expected value problem is solved and that solution is used to calculate each of the metrics in the full problem instance. A ‘*’ in an entry indicates a metric that was used as the objective function for the particular optimization model.
Table 3.5: Value of Stochastic Solution (VSS) and Expected Value of Perfect Information (EVPI) for Problem Instance A

<table>
<thead>
<tr>
<th>Metric</th>
<th>VSS</th>
<th>EVPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serviceability</td>
<td>0.34%</td>
<td>9.33%</td>
</tr>
<tr>
<td>Demand Satisfaction</td>
<td>12.3%</td>
<td>26.16%</td>
</tr>
</tbody>
</table>

Also, for problem instance A, Table 3.5 shows the VSS and EVPI for the two metrics, demand satisfaction and serviceability, that were chosen for optimization. The VSS when optimizing for serviceability is low, equal to 71.62% – 71.28% = 0.34%. The Wait–and–See serviceability shows that the expected value of perfect information is 9.33% = 80.95% – 71.62%, meaning that with perfect knowledge of what storm hazards will occur, the road system capacity could be increased by over 9%.

Though it took over 5.5 hours (Table 3.6), the extensive form of the demand satisfaction model was solved to optimality. The increased solution time was expected, because there are many more variables for the calculation of multicommodity flows in each hazard scenario. It should be noted that it took under an hour for the RP for demand satisfaction to be solved to within 1% of the optimal solution, and the other problems were rapidly solved. For the demand satisfaction metric, the VSS is 12.3%, and the EVPI is 26.16%. Finally, most of the models (except for the demand satisfaction model) were solved in under five seconds. CPLEX generates highly effective cuts for the serviceability optimization model, which aids in a rapid solution.

Table 3.7 shows the resilience metrics of Coastal City on problem instance B. Again, demand satisfaction was difficult to optimize, as optimizing for serviceability resulted in better demand satisfaction than the Benders decomposition approach and its variants did within the time constraint of twenty–four hours.

VSS and EVPI values for problem instance B are reported in Table 3.8. The VSS when optimizing for serviceability with a larger number of scenarios is larger, equal to 9.16% = 76.62% – 67.46%. This suggests that considering a greater number of scenarios improves the value of the stochastic programming approach, which is expected. The EVPI is 5.30% = 81.92% – 76.62%.
Table 3.6: Model Performance for Problem Instance A

<table>
<thead>
<tr>
<th>Model</th>
<th>Solution time* (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEV (Service)</td>
<td>&lt;1</td>
</tr>
<tr>
<td>EEV (Demand)</td>
<td>6.14</td>
</tr>
<tr>
<td>RP (Service)</td>
<td>0.62</td>
</tr>
<tr>
<td>RP (Demand)**</td>
<td>20752</td>
</tr>
<tr>
<td>WS (Service)</td>
<td>2.32</td>
</tr>
<tr>
<td>WS (Demand)</td>
<td>2.80</td>
</tr>
</tbody>
</table>

*times reported as wall clock time, not CPU time
**run on the cluster

Table 3.7: Metrics Results for Problem Instance B

<table>
<thead>
<tr>
<th>Model</th>
<th>Serviceability</th>
<th>Demand Satisfaction</th>
<th>Serviceability Recovery Time</th>
<th>Demand Satisfaction Recovery Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Status</td>
<td>65.36%</td>
<td>43.86%</td>
<td>37.81</td>
<td>37.92</td>
</tr>
<tr>
<td>EEV (Service)</td>
<td>*67.46%</td>
<td>47.88%</td>
<td>36.22</td>
<td>35.89</td>
</tr>
<tr>
<td>EEV (Demand)</td>
<td>66.99%</td>
<td>*46.90%</td>
<td>37.73</td>
<td>36.33</td>
</tr>
<tr>
<td>RP (Service)</td>
<td>*76.62%</td>
<td>66.09%</td>
<td>35.02</td>
<td>34.34</td>
</tr>
<tr>
<td>RP (Demand, Benders)</td>
<td>74.44%</td>
<td>*60.51%</td>
<td>35.94</td>
<td>35.11</td>
</tr>
<tr>
<td>RP (Demand, MW)</td>
<td>72.09%</td>
<td>*58.15%</td>
<td>37.30</td>
<td>37.09</td>
</tr>
<tr>
<td>RP (Demand, MW+)</td>
<td>71.70%</td>
<td>*57.67%</td>
<td>36.72</td>
<td>35.98</td>
</tr>
<tr>
<td>WS (Service)</td>
<td>81.92%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS (Demand)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*objective function in model

Table 3.8: Value of the Stochastic Solution (VSS) and Expected Value of Perfect Information (EVPI) for Problem Instance B

<table>
<thead>
<tr>
<th>Metric</th>
<th>VSS</th>
<th>EVPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serviceability</td>
<td>9.16%</td>
<td>5.30%</td>
</tr>
<tr>
<td>Demand Satisfaction (Benders)</td>
<td>≥13.61%</td>
<td>≤39.49%</td>
</tr>
<tr>
<td>Demand Satisfaction (MW)</td>
<td>≥11.25%</td>
<td>≤41.85%</td>
</tr>
<tr>
<td>Demand Satisfaction (MW+)</td>
<td>≥10.77%</td>
<td>≤42.33%</td>
</tr>
<tr>
<td>Demand Satisfaction (from Serviceability)</td>
<td>≥19.19%</td>
<td>≤33.91%</td>
</tr>
</tbody>
</table>
Benders decomposition provides at least an estimate of the VSS and EVPI for the demand satisfaction metric. From the standard Benders decomposition, EVPI is $\leq 39.49\%$, and VSS is $\geq 13.61\%$ according to the standard Benders decomposition, and results are similar for Benders decomposition with MW cuts and approximate core points. Adding the Mangnanti–Wong cuts with approximate core points did not produce better results, not even with the enhancement provided by Corollary 9 (MW+).

For the RP for serviceability, however, a better value for the demand satisfaction was obtained, since the optimal value of the demand satisfaction metric must be at least as high as the demand satisfaction found while optimizing serviceability. Here, VSS $\geq 19.19\%$ and EVPI $\leq 33.91\%$. The gap between the demand satisfaction metric when optimizing for serviceability and demand satisfaction demonstrates the oft–occurring slow convergence speed of Benders decomposition, as well as the general lack of conflict between the resilience metrics for this CIS and these hazard scenarios.

Table 3.9 reports the model solving time for problem instance $B$, with an additional breakdown between the master problem time and subproblem time for the Benders decomposition methods. The Benders decomposition algorithm always hit the time limit of twenty–four hours in all of its variants, while the other models generally were solved in less than twenty seconds. Optimizing for serviceability produced a better result than after twenty–four hours spent optimizing purely for demand satisfaction, both in terms of the quality of the solution and the time spent obtaining a solution.

While MW and MW+ methods were able to reduce the time spent solving the master problem, both spent more time solving the subproblems. The MW+ method actually spent more time on subproblems than on solving the master problem. These results indicate that the master problem is likely the computational bottleneck, but generating effective cuts for the master problem, as the Magnanti–Wong cuts seek to do, may require sufficient additional computational effort such that it outweighs the benefits of the cuts.
Table 3.9: Model Performance for Problem Instance B

<table>
<thead>
<tr>
<th>Model</th>
<th>Solution time* (sec.)</th>
<th>Master time* (sec.)</th>
<th>Subproblem time* (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEV (Service)</td>
<td>&lt;1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EEV (Demand)</td>
<td>3.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP (Service)</td>
<td>13.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP (Demand, Benders)**</td>
<td>86400+ (88794)</td>
<td>83010</td>
<td>5702</td>
</tr>
<tr>
<td>RP (Demand, MW)**</td>
<td>86400+ (86694)</td>
<td>73494</td>
<td>5544 + 7606 (MW)</td>
</tr>
<tr>
<td>RP (Demand, MW+)**</td>
<td>86400+ (87783)</td>
<td>37249</td>
<td>22480 + 27510 (MW)</td>
</tr>
<tr>
<td>WS (Service)</td>
<td>12.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS (Demand)</td>
<td>20.10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*times reported as wall clock time, not CPU time
**run on the cluster

3.6 Observations

Modeling and solving stochastic programming problems is a difficult problem even without integer variables; the addition of the integer variables complicates the solution of such problems. The problem instances in this paper are larger than most CIS modeling problems reported in the literature in terms of number of components and improvement decisions that can be made, and problem instance B contains more hazard scenarios than most efforts in the literature. The increased size of the problems made obtaining high-quality solutions for the demand satisfaction metric difficult, even when an improved Benders decomposition algorithm was applied. The serviceability metric, without any decomposition, was solved quickly even in the larger problem instance B, due to effective cuts based on the problem structure. Decomposing the demand satisfaction RP could have removed the ability to generate these effective cuts based on problem structure.

Even the limited solutions obtained here show potentially great value from solving these problems. For instance, the WS problems for demand satisfaction in Tables 3.4 and 3.7 both show that there is the possibility for "total resilience," because 100% of demand can be met in each hazard scenario with the right choice of improvements. While this could be due to the particular network and hazard scenarios, it demonstrates that it could be possible to mitigate or prevent much of the disruption that occurs due to natural hazards. Additionally, VSS increases from problem scenario A to B,
in which the number of scenarios increase. The increased VSS suggested that despite the increased computational difficulties from a larger number of scenarios, CIS planners can find greater benefit by considering a greater number of scenarios and potential events.

The results also indicated that improvements targeting either serviceability or demand satisfaction are ineffective at reducing the recovery time of the system. Further investigation will be required to verify whether this is due to the structure of the network, the effects of improvements, or the chosen recovery procedure.

Metric results in Tables 3.4 and 3.7 suggested, as was shown in Figure 2.6 (which graphed correlations between metrics) that the different resilience metrics do not conflict for the Coastal City CIS in these hazard scenarios. Future work could model other CIS systems, such as water distribution or electricity, to determine if this is due to the metrics or the modeling framework.

There are limitations to the work proposed for this chapter. The modeling framework does not allow for recourse decisions that change investment strategies. For instance, it could be useful to perform a combination of improvements and maintenance on a component of the CIS if it fails to fully recover or degrades from multiple events. If maintenance decisions were modeled as discrete variables, the current Benders decomposition strategy could not handle this extension.

The current modeling work has no ability to explicitly consider performance trade-offs, if any, between different solutions. An explicit multiobjective solution method could provide these trade-offs. A related issue is that parts of the problem may be unmodeled, or erroneously modeled, leading to optimal solutions for one metric performing poorly for the real-world system.

Finally, the solutions located by the optimization models are potentially sensitive to the hazard scenarios that were generated. Due to extreme variability and high uncertainty in forecasting future natural hazards, it is desirable that an optimization model locates a solution or solutions that can perform well for as wide a range of hazard scenarios as possible. The remainder of this dissertation develops an algorithm to attempt to address the issue of high uncertainty in future hazard scenarios.
4.1 Introduction

It is often difficult to directly implement solutions obtained from modeling and solving optimization-based engineering problems. Algorithmic techniques with theoretic guarantees of optimality as terminating conditions (e.g., mathematical programming) frequently find solutions located on the edge of feasible regions. Thus, uncertainties or unmodeled characteristics of an ill-posed problem may result in the obtained “optimal” solution to the model becoming inferior or even infeasible to decision-makers looking for solutions to the real system (Liebman, 1976). One approach to handle this issue is to use the optimization model to search for solutions that are maximally different in the decision space while being near-optimal (or optimal) (Brill, 1979). Such methods may provide high-quality and feasible alternatives from which decision-makers can identify potentially viable solutions or good starting points for manually creating solutions to the real problem (Brill et al., 1990). Numerous procedures for iteratively generating maximally different solutions using mathematical program-
ming methods have been reported in the literature (e.g., Abdulkadri and Ajibefun (1998); Baugh et al. (1997); Chang et al. (1982); Church et al. (1992); Lu and van Ittersum (2004); Mendoza et al. (1987)). Other methods have been proposed to generate maximally different solutions, such as evolutionary algorithms. The impetus for the use of evolutionary algorithms is that as population–based heuristics, these methods are well–suited to produce a set of nearly optimal solutions with diverse characteristics (Venkatasubramanian et al., 1995) and have been extended to find maximally different solutions (Loughlin et al., 2001). Decision–makers can use this set of solutions to explore the decision space and find a solution that satisfies unmodelled aspects of the problem.

In multiobjective optimization, solutions trade–off between the performances, or objective function values, of different solutions. Evolutionary algorithms are a popular method for approximating, if not explicitly locating, the Pareto front and the trade–offs among objectives (Coello Coello, 1999). A drawback to these methods is that most current evolutionary multiobjective optimization (EMO) algorithms concentrate on finding the best possible set of nondominated solutions with respect to the quantified objectives. The selection pressure to locate to nondominated solutions may cause the loss of high–quality, but dominated, solutions that are diverse in the decision space.

This paper develops the Evolutionary Multiobjective Optimization Algorithm for Diverse Solutions (EMODS), designed to simultaneously improve decision space diversity by retaining nearly–nondominated solutions diverse in the decision space, while approximating the Pareto front. Decision space diversity can be characterized by two aspects. The first, spread, is already commonly used for quantifying diversity obtained by EMO algorithms in the objective space, and occasionally decision space. The second aspect of decision diversity, distinctness of alternative solutions, is infrequently considered for multiobjective evolutionary algorithms. Both characterizations are crucial to fully assess the decision space diversity available in a solution set obtained for a multiobjective optimization problem.

The algorithm description begins with an outline of the overall algorithm. Next, a new and novel hypervolume–based fitness definition is proposed. A selection operator is described that is specific to this hypervolume–based fitness. The operator will select a slightly inferior solution as a parent of
a successive generation over a solution with higher fitness if the former solution is more diverse in
the decision space. An updating method is described for a diversity enhancing archive that retains a
target number of nondominated solutions and nearly–nondominated solutions with high diversity
in the decision space. The algorithm is benchmarked on a range of test problems with more metrics
for comparing diversity than other known efforts in the literature. Numerical experiments conducted
on test problems show that the proposed algorithm is competitive with, and frequently improves on,
other EMO algorithms intended to enhance decision space diversity.

4.2 Background

Necessary definitions and mathematical preliminaries follow. Let $X$ be an $n$–dimension space and
let the decision vector $\mathbf{x} = [x_1, \ldots, x_n] \in X$ be a solution to the following optimization problem:

$$\begin{align*}
\text{minimize} & \quad f(\mathbf{x}) = \langle f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}) \rangle \\
\text{subject to} & \quad \mathbf{x} \in X_F
\end{align*}$$

where $\mathbf{x} = [x_1, \ldots, x_n]$ and $X_F \subset X$ is the feasible region. Given two solutions $\mathbf{x}$ and $\mathbf{y}$, solution $\mathbf{x}$ is said
to dominate solution $\mathbf{y}$ if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, $\forall i \in \{1, \ldots, m\}$ and $\exists k \in \{1, \ldots, m\}$ such that $f_k(\mathbf{x}) < f_k(\mathbf{y})$; this is denoted as $\mathbf{x} \preceq \mathbf{y}$. Given a population of solutions, $P$, $\mathbf{x}$ is considered nondominated if $\{\mathbf{y} | \mathbf{y} \preceq \mathbf{x}, \mathbf{y} \in P\} = \emptyset$. For a population $P$, the set of nondominated solutions is called the nondominated
front. The set of all nondominated solutions in $X_F$ is known as the Pareto optimal set, defined as
$\{\mathbf{x} | \mathbf{x} \in X_F \text{ and } \nexists \mathbf{y} \in X_F \text{ where } \mathbf{y} \preceq \mathbf{x}\}$. The Pareto front is defined as the image of the Pareto set. In
this paper, solutions in $P$ whose objective values are near the nondominated front, but dominated,
will be called nearly–nondominated.

The algorithm is based on hypervolume, a measurement of the size of the space dominated by
a set of solutions with respect to a reference point that defines the extents of the region. Use of
hypervolume in multiobjective optimization began as a performance measure of algorithms (Zitzler
and Thiele, 1999), but now it is relatively common to use hypervolume (or something inspired
by it) within an evolutionary algorithm to help guide the search (Bader and Zitzler, 2011; Beume et al., 2007; Dorn, 2005). This is despite the high computational complexity of calculating the hypervolume (the best known algorithmic complexity for calculating $N$ points with $m$ objectives is $O(N \log N + N^{m/2})$ (Beume, 2009)), and other difficulties (#P-hardness (Bringmann and Friedrich, 2010)). Various algorithms have been proposed to achieve greater speed in calculating this quantity exactly (Beume, 2009; Fonseca and Manuel, 2006; While et al., 2012), and algorithms exist to approximate it as well (Bader et al., 2010; Ishibuchi et al., 2010). The calculation can also be sensitive to the selection of the reference point itself. In this paper, hypervolume calculations performed during the run of an algorithm assume normalized objective function values and an $m$–dimensional reference point $2.0^m$. Thus, the hypervolume of a set $A$ will be written as $Hyp(A)$, without a reference point.

4.2.1 Related Work

The pursuit of alternative solutions to improve the handling of unmodeled problem characteristics can be traced back to the modeling to generate alternatives (MGA) framework from Brill (1979). In this approach, inferior solutions within an allowed relaxation of the optimal solutions are systematically searched to locate alternatives with maximum diversity in the decision space. Numerous mathematical programming–based optimization problems have been used to illustrate the MGA method (Abdulkadri and Ajibefun, 1998; Baugh et al., 1997; Brill et al., 1990; Chang et al., 1982; Church et al., 1992; Lu and van Ittersum, 2004; Mendoza et al., 1987).

Solution diversity has been an aspect of EMO algorithms previously proposed. A distinction needs to be made, however, between diversity–driven and diversity–seeking EMO algorithms. The former use diversity in the decision space to improve convergence to the Pareto front and avoid local optima, as in Ursem (2002). Algorithms of the diversity–seeking type pursue diversity in solutions to provide distinct alternatives to decision–makers. Of course, diversity–driven EMO algorithms may produce alternative solutions. Analysis of these algorithms concentrates on the quality of the non-dominated set that has been obtained, rather than any investigation of the any alternatives.

It is comparatively recently that evolutionary algorithms have been designed to achieve decision
space diversity and discover alternative solutions. Li et al. (2002) created a niching genetic algorithm for single–objective problems to locate multiple optima and potentially novel alternative solutions. Species conservation was encouraged by the use of a species distance parameter that specified the furthest distance allowed between two solutions of the same species (i.e., niche). Species were adaptively defined according to the solutions with the best fitness and the species parameter. Additionally, Zechman and Ranjithan (2004) used niching via subpopulations to evolve and maintain solutions that are marginally sub–optimal with regard to a single–objective function, as well as maximally different in the decision space. Examples of other single–objective algorithms for diversity search exist in the literature (e.g., Caicedo (2010); Harrell (2001); Molina et al. (2010); Park and Ryu (2010); Ulrich and Thiele (2011)).

Some multiobjective evolutionary algorithms seeking to maximize decision space diversity also use niching mechanisms. For instance, Shir et al. (2009) proposed the Niching–CMA (Covariance Matrix Adaptation), which adds a niching mechanism to the multiobjective CMA–ES (Covariance Matrix Adaptation Evolutionary Strategy, Igel et al. (2007)). In Niching–CMA, selection is accomplished by nondominated sorting of solutions; then within each rank, niches are identified through consideration of aggregated objective and decision space distances between solutions. Thus, solutions similar in the objective and decision spaces are formed into niches, and each niche contributes to the subsequent generation.

An approach presented by Kramer and Danielsiek (2010), used evolutionary strategies and niching with two novel mechanisms for maintaining diversity in the objective space and decision space: rake selection (Kramer and Koch, 2009) and DBSCAN (Density Based Spatial Clustering of Applications with Noise, Ester et al. (1996)). Rake selection in two dimensions uses parallel lines to attempt to create a uniform approximation of the Pareto front. Niches are created by clustering using DBSCAN when a particular threshold measuring the need to recluster is exceeded. This threshold is the critical parameter for the algorithm, as it dictates exactly how niching works. The usual difficulties with niching strategies apply to these algorithms: determining the number of niches, and setting the niche radius. Density–based clustering such as DBSCAN removes the need to explicitly set these
parameters but requires its own parameters for the clustering process.

The Omni–Optimizer was proposed by Deb and Tiwari (2008) for single–objective and multiobjective optimization as an extension of the well–known NSGA–II algorithm. The Omni–Optimizer algorithm includes the means to allow more diverse solutions by relaxing dominance requirements through $\epsilon$–domination and sorting, and comparing decision space crowding in addition to the standard objective space crowding used in NSGA–II. Distinct from other efforts reviewed here, the intended purpose of the Omni–Optimizer for multiobjective optimization problems is not to locate alternative and potentially inferior solutions, but to find multiple global optima.

An algorithm proposed by Sarker and Ray (2009) is also based on NSGA–II. This algorithm ensures that solutions with maximum and minimum values in both the objective functions and decision variables are kept as elites. Additionally, after selecting the best objective solutions and solutions with largest and smallest values for each decision variable, more elites may be selected, if the number of elites is under a desired amount. If so, the remaining solutions are added iteratively, based on which solution has the best minimum Euclidean distance in the objective space. Nondominated sorting is also used by Toffolo and Benini (2003) in GDEA (Genetic Diversity Evolutionary Algorithm), which is built around GeDEM (Genetic Diversity Evaluation Method), where the definition of domination is extended to include decision space distances between solutions. Diversity in this algorithm is not pursued as a means to generate alternative solutions but instead as a method for improving convergence to the Pareto optimal set.

Diversity improving EAs based on assigning hypervolume–related fitness to solutions are reported in the literature. Ulrich et al. (2010b) proposed DIVA (Diversity Integrating Hypervolume–Based Search Algorithm), which uses a fitness assignment and selection procedure similar to that of the HypE algorithm (Hypervolume Estimation Algorithm for Multiobjective Optimization Bader and Zitzler (2011)), but the hypervolume of a solution is weighted by a desirability function based on the diversity of that solution. Diversity is measured by either the aggregated $n$–dimensional volume of hyperboxes in the decision space with a given size centered on the solutions in the current population or the sum of the distances to the centroid of the population.
Ulrich et al. (2010a) also presented the evolutionary algorithm DIOP (Diversity Integrating Multiobjective Optimizer), based on SPAM (Set Preference Algorithm for Multiobjective Optimization, Zitzler et al. (2010)). DIOP maximizes the weighted sum of two set measures: objective space hypervolume, and a diversity metric based on Solow–Polasky diversity (Solow and Polasky, 1994). A framework for the Solow–Polasky diversity metric was analyzed by Leinster (2010). DIOP can relax nondomination requirements to allow nearly–nondominated solutions to be included in the target population if they provide a sufficient increase in decision space diversity.

Ursem and Justesen (2012) presented an approach called the Multiobjective Distinct Candidates Optimization method to find a user–specified number of solutions to a multiobjective problem. This methodology uses a General Cluster–Forming Differential Evolution algorithm to pursue user–defined distinctness in the objective and decision spaces. The goal is to provide end–users of this approach with a limited and manageable set of possible solutions to consider trade–offs among fitness and solutions. This approach differs from the evolutionary algorithms above, which tend to select solutions based on fitness or decision diversity. Instead, the process employed is an on–line filtering of the solutions that is applied during the optimization process to obtain a final solution set with specific uniqueness requirements among its solutions.

### 4.2.2 Measuring Decision Space Diversity

When comparing the ability of different methods and algorithms to achieve decision space diversity, it rapidly becomes apparent that there is currently no consensus on a metric that accurately captures diversity. Despite known issues with pairwise distance metrics, most metrics are based on pairwise relationships among solutions. Others sidestep this difficulty by creating test problems with equivalent Pareto sets, as in Kramer and Danielsiek (2010), so analysis of a specific set can determine if decisions are well–spread throughout the decision space. In this section, the metrics, especially the pairwise distances, are reviewed, and new metrics are proposed to capture different aspects of diversity. Later in this paper, when evaluating algorithm performance on specific test problems, some problem–specific measures of diversity will be described.
A survey of diversity measures in single–objective optimization for real–valued decision vectors can be found in Corriveau et al. (2012). The same measures are used in multiobjective evolutionary algorithms generally and quantify the overall spread of solutions in the decision space. Unfortunately, these spread metrics fail to capture an important aspect of decision diversity for multiobjective optimization problems, which may be termed substitutability. A solution may be substituted for another (and chosen for implementation), if their objective function values are similar. The objective function of a single–objective problem is the basis for determining substitutability. If an algorithm finds many solutions that perform similarly to the best solution found, then they are substitutable and can be alternative solutions. If, in addition, these alternative solutions are diverse and distinct in the decision space, then the algorithm has generated alternative solutions. A measurement of the spread of the set of substitutable solutions in the decision space accurately captures the diversity in this case, because all solutions are substitutable for the optimal solution(s).

In contrast, for a multiobjective problem, a nearly–nondominated solution is unlikely to be substitutable for every nondominated solution. As a result, a nearly–nondominated solution will not be a good alternative for every nondominated solution. The lack of substitutability implies that a set of solutions that is well–spread in the decision space does not necessarily have viable, substitutable solutions. It is not argued here that having well–spread solutions is not important or useful. Rather, focusing exclusively on well–spread solutions can miss an important facet of solution diversity in multiobjective problems.

Despite this critical issue, various diversity metrics depending on pairwise decision space distances have been proposed in past studies. For instance, the average of all pairwise distances, normalized by the “diameter”, \( R \), of the decision space, was used by Shir et al. (2009). Given a population of solutions \( P \), then the Shir Diversity metric (\( SD \)) is calculated by:

\[
SD(P) = \frac{2}{R|P|(|P| - 1)} \cdot \sum_{\substack{x,y \in P \ x \neq y}} \|x - y\| \quad (4.1)
\]

This metric places a large emphasis on overall distances between solutions and can have high
values for sets that do not have a distribution approaching maximally disparate solutions.

The Solow–Polasky diversity metric, used in Ulrich et al. (2010a) and derived from Solow and Polasky (1994), is defined on a set of points as follows. Let set \( A = \{a_1, a_2, \ldots, a_n\} \) be given, and let \( d_{i,j} \) be the distance between \( a_i \) and \( a_j \) according to some distance metric. The Solow–Polasky metric for \( A \) is calculated by the expression

\[
SP(A) = e' S e
\]

where

\[
S = \begin{bmatrix}
    f(d_{1,1}) & f(d_{1,2}) & \cdots & f(d_{1,n}) \\
    f(d_{2,1}) & f(d_{2,2}) & \cdots & f(d_{2,n}) \\
    \vdots & \vdots & \ddots & \vdots \\
    f(d_{n,1}) & f(d_{n,2}) & \cdots & f(d_{n,n}) 
\end{bmatrix}^{-1},
\]

and \( e' = [1 \cdots 1] \) is an \( n \)-dimensional row vector. The function \( f \) must be such that \( f(0) = 1 \), \( f(\infty) = 0 \), and \( f'(x) \leq 0 \), \( \forall x \). Solow and Polasky (1994) recommended \( f = e^{-\theta d_{i,j}} \) with \( \theta > 0 \), where the parameter \( \theta \) must be supplied.

Leinster (2010) recently showed that this measurement is an instance of a concept known as magnitude of a metric space and provided a detailed mathematical framework for this concept. A view of measuring biodiversity of a population with the magnitude of the metric space can be found in Leinster and Cobbold (2012).

The Solow–Polasky diversity metric has the advantage of rewarding an algorithm that evenly spreads solutions over the decision space. Like any diversity metric based on all pairwise distances, it has some shortcomings when used to measure decision diversity in multiobjective problems. The metric does not address the issue of substitutability and can mischaracterize the diversity in a final solution set.
Thus, it is contended here that properly evaluating decision space diversity to account for both spread and the distinctness of alternative decisions requires multiple metrics. In the numerical experiments reported below in this paper, both the Shir Diversity metric and the Solow–Polasky metric are used. Three additional measures of diversity augment the characterization of decision space diversity. The first two measure different aspects of the worst–case spread.

Let \( d(\cdot, \cdot) \) be any distance metric; Euclidean distance is used in this paper. Define the *Average Minimum (AM)* of a population \( P \) as

\[
AM(P) = \frac{1}{|P|} \sum_{x \in P} d \left( x, \arg \min_{y \in P} d(x, y) \right)
\]

for all solutions. As an additional measure of spread, define the *Minimum Minimum (MM)* of a population \( P \) as

\[
MM(P) = \min_{x \in P} d \left( x, \arg \min_{y \in P} d(x, y) \right)
\]

These two metrics measure the average worst spread and worst spread among all solutions.

The final diversity measure defined is the *Average Distance from Nearest Objective (ADNO)*, which is mathematically expressed for a population \( P \) by

\[
ADNO(P) = \frac{1}{|P|} \sum_{x \in P} d \left( x, \arg \min_{y \in P} d(f(x), f(y)) \right)
\]

The *ADNO* metric differs from *AM* by calculating the decision space distance between a solution and its closest neighbor in the objective space. This metric tries to capture the distinctness of solutions that may be substituted for one another. Additionally, when a set has a larger value for *ADNO* than for *AM*, it shows that solutions near in the objective space generally have notable differences in the decision space.
ADNO concentrates on a particular aspect of diversity, the distinctness of alternative solutions, as do the other measures. If the user of the algorithm does not require a particular aspect of diversity in the final solution set, then there is no need to pick an algorithm that performs well on that metric. There are real-world problems, however, where knowledge of alternative solutions is valuable. For this reason, it is advocated that an algorithm seeking decision space diversity should be able to achieve both well-spread and maximally different decisions.

4.3 EMODS

EMODS defines novel mechanisms for its selection operator and archiving procedure. The algorithm takes four parameters. The size of the initial population, $t_p$ must be given. Two different targets for the size of an archive must be selected as well: $t_N$, the target number of nondominated solutions in the archive and $t_A$, the maximum desired size of the archive. The quantity $t_A - t_N$ is the maximum number of alternative solutions desired. The use of these parameters is discussed later. The final parameter needed is $r \in [0, 1]$, which determines the allowable relaxation for solutions to be considered nearly-nondominated. For example, $r = 0.85$ would permit solutions with fitness equal to 85% of the nondominated solutions to be considered nearly-nondominated. Pseudocode for the algorithm is outlined in Figure 4.1.

In each generation, the solutions in the current archive, $A$, are added to the population (Figure 4.1, Line 6). The hypervolume-based fitness, called hypervolume of the induced front and introduced in Section 4.4.1, is calculated for each solution in the combined population. A secondary measure of fitness, decision space crowding distance, is calculated to rank solutions by decision space diversity. After fitness evaluations, parents are selected according to the selection operator in Section 4.4.2. The selection operator chooses solutions with similar induced front hypervolume on the basis of their decision diversity fitness. The archive update procedure, which chooses nondominated and nearly-nondominated solutions from the population, is described in Section 4.5. Recombination and mutation occurs before the current population is entirely replaced by the offspring. In the numerical experiments in this paper these operators are the simulated binary crossover and polyno-
**Require:** population size parameters \( t_P, t_A, t_N \); relaxation parameter \( r \)

1: Initialize population \( P, |P| = t_P \) and archive \( A = \emptyset \)

2: \( \text{iter} \leftarrow 0 \)

3: Calculate objective function values \( \forall x \in P \)

4: \( \text{while} \) \( \text{iter} < \text{max}\_\text{iter} \) \( \text{do} \)

5: \( \text{iter} \leftarrow \text{iter} + 1 \)

6: \( P = P \cup A \) \( \triangleright \) Figure 4.3

7: \( h_F = \text{Hyp}(F_x) \), \( \forall x \in P \)

8: \( h_F = \text{Hyp}(F) \) where \( F \) is the nondominated front of \( P \)

9: Decision space crowding

10: \( P' = \text{Select}(P, r) \) \( \triangleright \) Figure 4.4

11: \( A = \text{ArchiveUpdate}(P, h_F, t_A, t_N) \)

12: \( P = \text{RecombineMutate}(P') \)

13: Calculate objective function values \( \forall x \in P \)

14: \( \text{end while} \)

15: \( P = P \cup A \)

16: Calculate \( \text{Hyp}(F_x), \forall x \in P \)

17: \( h_F = \text{Hyp}(F) \) where \( F \) is the nondominated front of \( P \)

18: \( A = \text{ArchiveUpdate}(P, h_F, t_A, t_N) \)

19: \( \text{return} A \)

Figure 4.1: Pseudocode for EMODS

Mial mutation operators (Deb and Tiwari, 2008), both of which are modified to keep solutions within the allowed range for each decision variable (i.e., satisfy box constraints).

### 4.4 Hypervolume Fitness and Selection

A component of EMODS is a novel hypervolume–based fitness that scalarizes the fitness of solutions to rank a population. Using a single value to represent a solution’s fitness helps to evaluate and compare dominated solutions that may have dramatically different objective function values.

#### 4.4.1 Hypervolume–based Fitness

The fitness of a solution is evaluated based on what is termed an *induced* front, which is a potential nondominated front that would result if a given solution is included in the set of nondominated solution. An induced front is formally defined as follows.


**Definition 2** (Induced Front). *If \( F \) is the set of nondominated solutions in a population of solutions, \( P \), let \( D_x = \{ p \in F \mid \mathbf{p} \preceq x \} \) be the solutions in \( F \) that dominate \( x \in P \). Then the set \( F_x = (F \setminus D_x) \cup \{x\} \) is defined as the induced front of \( x \).*

It follows that for solutions \( x \in F \), \( D_x = \emptyset \), and \( F_x = F \). In short, the induced front is the nondominated front formed if the nondominated solutions that dominate \( x \) are temporarily removed from the population. For each solution, the hypervolume of the induced front is calculated and assigned as the solution's fitness. Induced front hypervolume for \( x \) is written as \( h_x = Hy_p(F_x) \). In Figure 4.2, the induced front hypervolume of solution \( a \) is the area of the shaded region, representing the hypervolume of the induced front of \( a \), which is formed by removing solutions \( b \) and \( c \) from the population. Infeasible solutions are assumed to have a fitness of 0, unless the use of a penalty function is desired.

Calculating the induced front for each solution can be accomplished by using a variant on a nondominated sort. An algorithm for calculating induced front hypervolume for a population is given in Figure 4.3. The algorithm first conducts a variant on a fast nondominated sort in Lines 4–13. For each solution \( x \), \( D_x \) is tracked (Figure 4.3, Line 6), which adds no complexity to the sorting. At the end of the sort, the current Pareto front approximation is used to create the set of solutions dominating \( x \), so \( D_x = D_x \cap F \) (Figure 4.3, Line 15). Then, for each solution \( x \in P \), \( Hy_p(F_x) \) is calculated to obtain \( h_x \) (Figure 4.3, Line 21).
1: procedure INDUCEDFRONTFITNESS(population P)
2: \[ D_x = \emptyset, \quad \forall x \in P \quad \text{\textup{\texttt{\color{red}{\triangleright}} solutions dominating } x} \]
3: \[ F = \emptyset \quad \text{\textup{\texttt{\color{red}{\triangleright}} nondominated solutions} \]
4: \textbf{for} \ x \in P \ \textbf{do}
5: \quad \textbf{for} \ y \in P; y \neq x \ \textbf{do}
6: \quad \quad \textbf{if} \ y \preceq x \ \textbf{then}
7: \quad \quad \quad D_x = D_x \cup \{y\}
8: \quad \quad \textbf{end if}
9: \quad \textbf{end for}
10: \quad \textbf{if} \ D_x = \emptyset \ \textbf{then}
11: \quad \quad F = F \cup \{x\}
12: \quad \textbf{end if}
13: \textbf{end for}
14: \textbf{for} \ x \in P \ \textbf{do}
15: \quad D_x = D_x \cap F
16: \textbf{end for}
17: \text{normalize all objective function values}
18: \[ h_F = Hy p(F) \]
19: \textbf{for} \ x \in P \ \textbf{do}
20: \quad \textbf{if} \ x \notin F \ \textbf{then}
21: \quad \quad h_x = Hy p(F_x)
22: \quad \textbf{else}
23: \quad \quad h_x = h_F
24: \quad \textbf{end if}
25: \textbf{end for}
26: \textbf{end procedure}

Figure 4.3: Pseudocode for Induced Front Fitness Calculation
4.4.2 Selection Operator

The selection operator is a binary tournament and compares two solutions according to the following criteria in order: feasibility, induced front hypervolume, and decision diversity. A clear winner in any criterion means that the remaining criteria are not checked. Feasibility checking will be problem dependent. A possible scheme is to choose a feasible solution over an infeasible solution and if both solutions are infeasible, choose the solution with the smallest sum of constraint violations.

If both solutions chosen for the binary tournament are feasible, the induced front hypervolume is used to choose the winner. Given two feasible solutions \( x \) and \( y \), the user-specified relaxation parameter \( r \) dictates that for \( x \) to be chosen, it must be that

\[
h_y < r \cdot h_x
\]

where \( h_x \) is the induced front hypervolume of \( x \). Otherwise, if

\[
r \cdot h_x \leq h_y \leq h_x
\]

then the final criterion, the decision diversity, is used to pick the winner.

Decision diversity was evaluated using a decision space crowding distance operator. It is calculated similarly to the procedure described in Deb and Tiwari (2008), except that solutions with variables at the extremes of the population receive infinity as their decision crowding distance, as in Sarker and Ray (2009). For each decision variable, the population \( P \) is sorted in ascending order, so that for the \( k^{th} \) decision variable, \( P = \{x_{k(1)}, x_{k(2)}, \ldots, x_{k(n)}\} \) is increasing in the \( k^{th} \) decision variable. For \( i = 2, \ldots, n - 1 \), given a solution \( x_{k(i)} = [x_{k(i),1}, x_{k(i),2}, \ldots, x_{k(i),n}] \), the decision crowding distance is calculated variable-wise as

\[
d_{c_k}(x_{k(i)}) = \frac{x_{k(i+1),k} - x_{k(i-1),k}}{x_{k(n),k} - x_{k(1),k}}
\]
and for a solution $x$ as

$$d_c(x) = \sum_{k=1}^{n} d_{c_k}(x).$$

(4.8)

Other decision diversity measures can be used for the secondary fitness calculation. In the main algorithm, decision diversity is calculated in Line 9 of Figure 4.1.

The selection procedure chooses parent solutions until $t_P$ parents have been chosen for recombination and mutation. The children of these parents replace the entire population, resetting the size of the population to $t_P$, as the population size may have increased when the archive solutions were combined with the population.

### 4.5 Archive

EMODS introduces a diversity-maintaining archive to hold nondominated solutions while retaining alternative, nondominated and nearly-nondominated solutions that are diverse in the decision space. Solutions in the archive set participate in the fitness calculation and selection process, and the archive is updated every generation. The archive, $A$, contains set of nondominated solutions, $N_A$, and a set of diversity solutions, $D_A$ such that $A = N_A \cup D_A$, and $N_A \cap D_A = \emptyset$. Solution diversity is maintained by determining if the decision variables of a solution $x$ violate minimum allowed distances, $\epsilon$, between the decision variables of another solution. If so, then the solutions are said to overlap. A formal definition follows.

**Definition 3 (Overlap).** Let $n$-dimensional points $x, y, \epsilon$ be given. Let also $d(\cdot, \cdot)$ be a distance metric, or set of metrics, such that $d(x_i, y_i)$ can be calculated $\forall i \in \{1, \ldots, n\}$. Points $x$ and $y$ overlap if $\forall i \in \{1, \ldots, n\}, d(x_i, y_i) \leq \epsilon_i$.

Archive solutions are not permitted to overlap. If a pair of solutions overlaps, then one will be removed from the archive. The induced front hypervolumes of the solutions are compared, and the solution with lower induced front hypervolume is removed. Another view on overlapping is to envi-
1: procedure ARCHIVEUPDATE(population $P$, fitness target $h_F$, size targets $t_A$, $t_N$) 
2: Calculate standard deviations $s$, scale factor $\alpha$ 
3: Update $\epsilon = \alpha s$ 
4: $N_A = \{x | x \in P, h_x = h_F\}$ 
5: if $|N_A| > t_N$ then 
6: $R = \text{trim\_nondominated}(N_A)$ 
7: end if 
8: $D_A = R \cup \{x | x \in P, r \cdot h_F \leq h_x < h_F\}$ 
9: for $x \in D_A$ do 
10: $O = \{y | y \in A \text{ and } x, y \text{ Overlap for } \epsilon\}$ 
11: if $h_x > h_y, \forall y \in O$ then 
12: $D_A = D_A \setminus O$ 
13: else 
14: $D_A = D_A \setminus \{x\}$ 
15: end if 
16: end for 
17: return $A = N_A \cup D_A$ 
18: end procedure

Figure 4.4: Pseudocode of Archive Update Procedure

A solution $x$ has a hyperbox with sides of length $[2\epsilon_1 \ \ldots \ 2\epsilon_n]$ centered on $x$. A solution $y$ overlaps $x$ if $y$ is inside the hyperbox described, as this would mean that the distances between all $x_i$ and $y_i$ are less than or equal to $\epsilon_i$. The vector $\epsilon$ is adaptively determined, based on the distribution of the current population’s decision variables and the parameter $t_A$, the maximum desired archive size.

Pseudocode for the archive update procedure is given in Figure 4.4, and the procedure is described in detail below.

A target fitness value, $h_F$, is obtained from the hypervolume of the current population’s non-dominated front, $F$. The target $h_F$ is used to determine which solutions will be in $N_A$ and which are eligible to be in $D_A$. After obtaining the target $h_F$, $N_A$ is set equal to $F$.

The desired size of $N_A$ is user-specified and given by the parameter $t_N$. When the procedure results in a set $N_A$ such that $|N_A| > t_N$, then $N_A$ is trimmed until its size reaches $t_N$. In EMODS, trimming is accomplished by removing solutions from $N_A$ with the lowest hypervolume contributions (Line 6, Figure 4.4). Solutions removed from $N_A$ are added to $D_A$ in Line 8.
The diversity mechanism in the archive set uses the concept of overlapping solutions from Definition 3. The vector $\varepsilon$ defines as the minimum decision space distances allowed between a solution in $D_A$ and another in the archive. The process of determining $\varepsilon$ is motivated by the observation that the population and archive of each generation will be in a particular region of the decision space, which may be called the active region. As the algorithm has a maximum desired size of the archive, $t_A$, the values of $\varepsilon$ should be sufficiently large so that at most $t_A$ solutions can fit into the current active region. The tightest packing of the solutions possible, considering that the hyperboxes centered on each solution that disallow overlapping, would be a grid, where each grid point is a solution.

The basis for the relative lengths of each side of the hyperbox are determined by the standard deviation, $s_i$, of the decision variables individually (e.g., $s_1$ is the standard deviation of the values of $x_1$ currently in the population). A scale factor, $\alpha$, needs to be determined for the standard deviations such that $\varepsilon = \alpha s_i$ will allow at most $t_A$ solutions into the archive. Let $l_i$ be the difference between the largest and smallest value of $x_i$ in $A \cup P$ and consider $l_i$ as the length of each side of the current active region. Along each dimension, at most $\left\lfloor \frac{l_i}{\alpha s_i} \right\rfloor$ solutions can be admitted. For the current active region, a total of $\prod_{i=1}^{n} \left\lfloor \frac{l_i}{\alpha s_i} \right\rfloor$ solutions can fit. To satisfy the limit given by $t_A$, it must be that

$$\prod_{i=1}^{n} \left\lfloor \frac{l_i}{\alpha s_i} \right\rfloor \leq t_A.$$  \hspace{1cm} (4.9)

To more easily solve this equation for $\alpha$, an approximation is used, and the floor function is dropped. Thus, the scale factor $\alpha$ is given as

$$\alpha = \left( \frac{1}{t_A} \frac{1}{\prod_{i=1}^{n} s_i} \right)^{\frac{1}{2}}.$$  \hspace{1cm} (4.10)

The approximation unfortunately permits $t_A$ to become a soft limit, but the later numerical experiments show that this does not usually cause a problem.

For each solution $x$ in $D_A$, a set $O$ is calculated that contains the solutions in $A$ that overlap $x$ considering the $\varepsilon$ values calculated for the current generation (Figure 4.4, Line 10). For $x$ to remain in the archive, $h_x$ must be greater than $h_y$ for all solutions $y \in O$. If this is the case, all solutions in $O$
are removed from $A$. Otherwise, $x$ is dropped from the $D_A$. An immediate consequence is if $x \in D_A$ overlaps with a solution in $N_A$, $x$ is always removed.

A possible scenario for the updates is illustrated in Figure 4.5, where hyperboxes are shaded to illustrate the areas where no overlap may be permitted for diversity solutions. Diversity solutions are represented by diamonds and nondominated solutions by circles. Potential diversity solutions $a$ and $b$ overlap in Figure 4.5. This means that $h_a$ and $h_b$ are compared to see which is greater. Additionally in this figure, the potential diversity solution $d$ overlaps with the nondominated solution $c$, so $d$ will not be allowed into the archive.

### 4.6 Numerical Experiments

EMODS was tested on six problems with two and three objectives whose Pareto fronts are diverse in the decision space due to either multiple globally optimal solutions or local fronts with near–Pareto optimal solutions. The two objective problems were also chosen because other diversity enhancing EMO algorithms were tested on these problems. Table 4.1 summarizes the test problem characteristics.
<table>
<thead>
<tr>
<th>Problem Name</th>
<th>Objectives/Variables</th>
<th>Objective Functions (all minimizations)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two–On-One</td>
<td>2/2</td>
<td>$f_1(x) = x_1^4 + x_2^4 + x_1^2 + x_2^2 - 10x_1x_2 + 0.25x_1 + 20$</td>
<td>Global and local front</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(x) = x_1^2 + x_2^2$</td>
<td></td>
</tr>
<tr>
<td>(Preuss et al., 2006)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Omni–Test (Deb and Tiwari, 2008)</td>
<td>2/5</td>
<td>$f_1(x) = \sum_{i=1}^{5} \sin(\pi x_i)$</td>
<td>243 clusters of global optima</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(x) = \sum_{i=1}^{5} \cos(\pi x_i)$</td>
<td></td>
</tr>
<tr>
<td>Lamé Spheres (Emmerich and Deutz, 2007; Shir et al., 2009)</td>
<td>2/4</td>
<td>$f_1(x) = \cos(x_1) \left(1 + \sin^2 g(x)\right)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(x) = \sin(x_1) \left(1 + \sin^2 g(x)\right)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(g(x) = \frac{\pi}{4} \sum_{i=2}^{4} x_i)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_i \in [0, 1], x_j \in [1, 5], \forall i = 2, \ldots, 4$</td>
<td></td>
</tr>
<tr>
<td>EBN (Emmerich et al., 2005)</td>
<td>2/10</td>
<td>$f_1(x) = \left(\sum_{i=1}^{10}</td>
<td>x_i</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(x) = \left(\sum_{i=1}^{10}</td>
<td>x_i - 1</td>
</tr>
<tr>
<td>Lamé Scaled (Emmerich and Deutz, 2007; Shir et al., 2009)</td>
<td>3/7</td>
<td>$f_1(x) = \cos(x_1) \left(1 + \sin^2 g(x)\right)$</td>
<td>Lamé Superspheres scaled up in variables and objectives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(x) = \sin(x_1) \cos(x_2) \left(1 + \sin^2 g(x)\right)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_3(x) = \sin(x_1) \sin(x_2) \left(1 + \sin^2 g(x)\right)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(g(x) = \frac{\pi}{8} \sum_{i=3}^{7} x_i)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_1, x_2 \in [0, \frac{\pi}{2}], x_i \in [1, 5], \forall i = 3, \ldots, 7$</td>
<td></td>
</tr>
<tr>
<td>DTLZ2 (Deb et al., 2005)</td>
<td>3/7</td>
<td>$f_1(x) = (1 + g(x))\cos(x_1 \pi/2)\cos(x_2 \pi/2))$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(x) = (1 + g(x))\cos(x_1 \pi/2)\sin(x_2 \pi/2))$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_3(x) = (1 + g(x))\sin(x_1 \pi/2))$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(g(x) = \sum_{i=3}^{7} (x_i - 0.5)^2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$x_i \in [0, 1], \forall i$</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.2: Parameters for Metric Calculations

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>Reference Point</th>
<th>Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-on-One</td>
<td>[35,7]</td>
<td>8.4853</td>
</tr>
<tr>
<td>Omni-Test</td>
<td>[1,1]</td>
<td>13.4164</td>
</tr>
<tr>
<td>Lamé Superspheres</td>
<td>[2,2]</td>
<td>7.1040</td>
</tr>
<tr>
<td>EBN</td>
<td>[2,2]</td>
<td>3.1623</td>
</tr>
<tr>
<td>LaméScaled</td>
<td>[2,2,2]</td>
<td>2.6457</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>[2,2,2]</td>
<td>2.6457</td>
</tr>
</tbody>
</table>

4.6.1 Metrics

The ability of EMODS to approximate the Pareto front will be measured by the hypervolume metric. For assessing the diversity of the final solution sets, a variety of metrics will be used. As previously argued in Section 4.2.2, measuring diversity requires consideration of the spread in the decision space and the distinctness of alternative solutions. The Solow–Polasky metric and the Shir metric (pairwise distances) represent the overall spread of the solutions in the decision space. Additionally, the metrics proposed earlier, $AM$, $MM$ and $ADNO$, will be calculated for each problem, the last representing the distinctness of solutions substitutable in their objective value functions. When results of these diversity metrics for other algorithms have been reported, they are compared with the results of EMODS. Note that these metrics are only calculated for a set of nearly–nondominated or nondominated solutions obtained at the end of the run, not for the entire final population.

Table 4.2 shows the parameter values used to calculate the metrics, such as the reference point for final hypervolumes, and the diameter for the Shir diversity metric. Objective function values were not normalized for this evaluation, as they were for induced front hypervolume calculations. For the Solow–Polasky metric, a value of $\theta = 10$ was used for each test problem.

4.6.2 Final Solution Set Selection

The final archive set obtained from EMODS consists of nondominated and nearly–nondominated solutions to help decision–makers explore the solution space. As the final archive set may be large,
it could be helpful to keep the explored set managably small to avoid inundating a decision–maker with too many choices. Additionally, metrics such as the Solow–Polasky diversity metric and hypervolume monotonically increase with the number of unique solutions in a set (with the qualifier that solutions must be nondominated for hypervolume). Thus, while results will be reported for the obtained archives, an equitable comparison with other published results requires sets with the same number of solutions. As the maximum desired archive size is a soft limit, a method for trimming final archives after the run to some target size is necessary for appropriate comparisons.

Metrics will be calculated for the final archive set $A$ and denoted as $EMODS$, as well as for set $N_A$, contained in the archive $A$. The metric calculations for $N_A$ are reported as $EMODS–N_A$ in the results. Then, an algorithm selects a final diversity set, a mixture of nondominated and nearly–nondominated solutions in $A$. The final diversity set is reported as $EMODS$ Nearly–Nondominated Set ($EMODS$–NN) in the results. This method tries to ensure that the nondominated front found is well–represented, but exchanges a solution from $N_A$ in favor of an alternative solution, if the alternative solution is more diverse (in the sense of being more distant from other solutions in the decision space).

The nearly–nondominated solutions in $D_A$ are used to pick the elements for the $EMODS$–NN. Define $MD(x)$ as the decision space distance from $x$ to the nearest solution in the archive. For each solution $x \in N_A$, $MD(x)$ is compared against $MD(y)$ if $y$ is the $i^{th}$ nearest objective space neighbor of $x$. In essence, this compares $x$ against the $k^{th}$ most substitutable solution for $x$ in $A$. If the $k^{th}$ nearest objective space neighbor is already in $EMODS$–NN, $k$ is increased by 1. In the numerical experiments conducted in this paper, five was the highest value checked for $k$ (i.e., $k = 5$ in Figure 4.6). Raising this threshold often resulted in higher diversity and lower hypervolume (but not uniformly). This trimming procedure was only conducted as post–analysis to provide an equitable comparison.

4.6.3 Results

$EMODS$ was implemented in Python using the DEAP package (De Rainville et al., 2012). Hypervolumes were calculated via a Python wrapper created by SWIG for the C code available from http://www.wfg.csse.uwa.edu.au/ that implements the hypervolume calculation algorithm presented
1: **procedure** NEARLY—NONDOMINATEDSET(nondominated set \( N_A \), set of alternatives \( D_A \), limit \( k \))

2: \( F = \emptyset \)

3: for \( x \in N_A \) do

4: \( i \rightarrow 1 \)

5: \( y = i^{th} \) nearest solution to \( x \) in decision space

6: while \( y \in F \) do

7: \( i \rightarrow i + 1 \)

8: if \( i \geq k \) then

9: \( F = F \cup \{x\} \)

10: continue to next \( x \in N_A \)

11: else

12: \( y = i^{th} \) nearest solution to \( x \) in decision space

13: end if

14: end while

15: if \( MD(x) > MD(y) \) then

16: \( F = F \cup \{x\} \)

17: else

18: \( F = F \cup \{y\} \)

19: end if

20: end for

21: return \( F \)

22: **end procedure**

Figure 4.6: Pseudocode for Final Near–Pareto Set Selection
Table 4.3: Algorithm Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Runs</td>
<td>30</td>
</tr>
<tr>
<td>Generations</td>
<td>200</td>
</tr>
<tr>
<td>Population Size, ( t_P )</td>
<td>50</td>
</tr>
<tr>
<td>Set ( N_A ) target size, ( t_N )</td>
<td>50</td>
</tr>
<tr>
<td>Archive target size, ( t_A )</td>
<td>200</td>
</tr>
<tr>
<td>Crossover Parameter</td>
<td>15</td>
</tr>
<tr>
<td>Crossover Probability</td>
<td>1.0</td>
</tr>
<tr>
<td>Mutation Parameter</td>
<td>20</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>( 1/n )</td>
</tr>
<tr>
<td>Relaxation, ( r )</td>
<td>0.99</td>
</tr>
</tbody>
</table>

by While et al. (2012). All source code is available upon request.

The results from EMODS are compared to those reported by the N–CMA (Shir et al., 2009) and DIOP (Ulrich et al., 2010a), two evolutionary algorithms representative of diversity–enhancing algorithm in literature. Both algorithms reported results for populations of size 50. Where possible, all metrics from other studies are compared, though this is not possible for the new metrics presented in this paper. The target size parameters \( t_P \), \( t_A \), and \( t_N \) were set to 50, 200, and 50, respectively. The latter two parameters imply that the algorithm is looking for approximately 150 alternative solutions. Table 4.3 provides a summary of the EMODS parameter values used in the tests. The metric results reported for EMODS are for population sizes of 50, and the algorithm in Figure 4.6 was used to trim EMODS final archive size to 50 when necessary.

The results of the runs are summarized in Tables 4.4 through 4.10. The results reported in this paper are from thirty runs with 10,000 function evaluations on each test problem, as in the experiments reported in Shir et al. (2009). Results reported by Ulrich et al. (2010a) were obtained with 100,000 function evaluations. The performance of EMODS with 10,000 function evaluations is nevertheless competitive.

The results of hypervolumes obtained (Table 4.4) show that EMODS achieves competitive or slightly better hypervolume when compared to N–CMA. Compared to DIOP’s hypervolume on the Omni–Test problem EMODS did well against a set of runs when diversity was considered (without di-
Table 4.4: Hypervolume Results (* indicates values unavailable from other papers, bold indicates the best score on that test problem)

<table>
<thead>
<tr>
<th></th>
<th>Two–on–One</th>
<th>Omni–Test</th>
<th>Lamé</th>
<th>EBN</th>
<th>LaméScaled</th>
<th>DTLZ2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMODS</strong></td>
<td>Mean</td>
<td>174.5</td>
<td>30.28</td>
<td>3.208</td>
<td>3.316</td>
<td>7.403</td>
</tr>
<tr>
<td></td>
<td>Std. Dev.</td>
<td>0.006</td>
<td>0.0507</td>
<td>1 e-4</td>
<td>0.027</td>
<td>9 e-4</td>
</tr>
<tr>
<td><strong>EMODS–NA</strong></td>
<td>Mean</td>
<td>174.5</td>
<td>30.25</td>
<td>3.207</td>
<td>3.311</td>
<td>7.399</td>
</tr>
<tr>
<td></td>
<td>Std. Dev.</td>
<td>0.006</td>
<td>0.0509</td>
<td>5 e-5</td>
<td>0.027</td>
<td>8 e-4</td>
</tr>
<tr>
<td><strong>EMODS–NN</strong></td>
<td>Mean</td>
<td>172.7</td>
<td>30.00</td>
<td>3.200</td>
<td>3.237</td>
<td>7.357</td>
</tr>
<tr>
<td></td>
<td>Std. Dev.</td>
<td>0.9174</td>
<td>0.0865</td>
<td>4 e-3</td>
<td>0.0468</td>
<td>0.010</td>
</tr>
<tr>
<td><strong>N–CMA</strong></td>
<td>Mean</td>
<td>173.44</td>
<td>30.27</td>
<td>3.172</td>
<td>3.295</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Std. Dev.</td>
<td>0.14</td>
<td>0.05</td>
<td>0.037</td>
<td>0.038</td>
<td>*</td>
</tr>
<tr>
<td><strong>DIOP</strong> (w_o = 0.91)</td>
<td>Mean</td>
<td>*</td>
<td>30.25</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Std. Dev.</td>
<td>*</td>
<td>0.03</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

versity considerations, DIOP achieved slightly higher hypervolume). There is some loss of optimality on the Omni–Test problem when considering the EMODS–NN obtained from the archive; however, this is at least in part a result of fewer function evaluations used for EMODS runs. Thirty test runs of EMODS with 50,000 function evaluations on the Omni–Test problem obtained an average hypervolume of 30.24 ± 0.06 for the EMODS–NN and higher values for the other final sets. The hypervolume of the EMODS–NN with 10,000 function evaluations is within 99% of the hypervolume of the non-dominated set obtained by the algorithm, as intended by the relaxation parameter. On the other test problems, the hypervolume of EMODS was competitive or better (Table 4.4).

EMODS performs competitively on the Solow–Polasky and Shir Diversity metrics, both of which represent overall spread in the decision space. As noted previously, the Solow–Polasky metric monotonically increases with the number of points in the set. Table 4.5 omits reporting the results for the final archive from EMODS, because the archive in EMODS has no strict size limit and comparisons would be unequal. Table 4.5 shows that for Omni–Test, EMODS–NA has a lower Solow–Polasky value than DIOP, however, the EMODS–NN has a higher average Solow–Polasky value and is nearly as robust as DIOP. The Omni–Test problem has a specific structure enabling use of a problem–specific diversity measure, namely, that there are 243 clusters of Pareto optimal solutions in the decision
Table 4.5: Solow–Polasky Results (* indicates values unavailable from other papers, bold indicates the best score on that test problem)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Two–on–One</th>
<th>Omni–Test</th>
<th>Lamé</th>
<th>EBN</th>
<th>LaméScaled</th>
<th>DTLZ2</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMODS–NA</td>
<td>Mean</td>
<td>11.87</td>
<td>42.33</td>
<td>40.78</td>
<td>48.84</td>
<td>45.51</td>
</tr>
<tr>
<td></td>
<td>Std. Dev.</td>
<td>0.09</td>
<td>2.08</td>
<td>2.24</td>
<td>0.43</td>
<td>1.15</td>
</tr>
<tr>
<td>EMODS–NN</td>
<td>Mean</td>
<td>26.06</td>
<td>49.89</td>
<td>46.42</td>
<td>49.83</td>
<td>49.19</td>
</tr>
<tr>
<td></td>
<td>Std. Dev.</td>
<td>1.38</td>
<td>0.26</td>
<td>1.29</td>
<td>0.20</td>
<td>0.49</td>
</tr>
<tr>
<td>DIOP</td>
<td>Mean</td>
<td>*</td>
<td>48.7</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>(w₀ = 0.91)</td>
<td>Std. Dev.</td>
<td>*</td>
<td>0.9</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 4.6: Omni-Test Clusters Found

<table>
<thead>
<tr>
<th>Algorithm/Final Set</th>
<th>Clusters Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMODS</td>
<td>185.43 ± 43.71</td>
</tr>
<tr>
<td>EMODS–NA</td>
<td>37.5 ± 2.99</td>
</tr>
<tr>
<td>EMODS–NN</td>
<td>48.87 ± 0.99</td>
</tr>
<tr>
<td>DIOP (w₀ = 0.91)</td>
<td>39.9 ± 3.7</td>
</tr>
</tbody>
</table>

space. EMODS–NN obtains a greater number of clusters in the decision space (Table 4.6) without a corresponding increase in the Solow–Polasky metric. The absence of an associated increase suggests that the Solow–Polasky metric does not always directly correspond to the number of different regions found in the decision space. The spread of clusters in the decision space for a run of EMODS can be seen in Figure 4.7.

The issues with pairwise distance metrics become apparent in the results for the Shir Diversity (Table 4.7). In the Omni-Test, problem DIOP reports an average score of 0.66, significantly better than EMODS and N–CMA. The discrepancy between the number of clusters found and the spread metrics provides experimental evidence that having well-spread solutions in the decision space is not sufficient to guarantee the presence of substitutable alternative solutions.

It is worth noting that for most of the test problems in Table 4.5, EMODS–NN has a Solow–Polasky score near 50, indicating distinct solutions were identified. The exception is for the Two–on–One problem, which is not a multi–global problem but instead has a local and a global Pareto front. Fig-
Figure 4.7: Sample of solution clusters found by EMODS for Omni–Test problem
Table 4.7: Shir Diversity Results (* indicates values unavailable from other papers, ** indicates the best score on that test problem)

<table>
<thead>
<tr>
<th></th>
<th>Two–on–One</th>
<th>Omni–Test</th>
<th>Lamé</th>
<th>EBN</th>
<th>LaméScaled</th>
<th>DTLZ2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMODS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.1413</td>
<td>0.3800</td>
<td>0.4579</td>
<td>0.5538</td>
<td>0.4391</td>
<td>0.4792</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.007</td>
<td><strong>0.015</strong></td>
<td>0.012</td>
<td>0.0038</td>
<td><strong>0.0078</strong></td>
<td>0.0148</td>
</tr>
<tr>
<td><strong>EMODS–NN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.0756</td>
<td>0.3708</td>
<td>0.4078</td>
<td>0.5508</td>
<td>0.3707</td>
<td>0.2272</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td><strong>9e-4</strong></td>
<td>0.0200</td>
<td>0.0204</td>
<td>0.0067</td>
<td>0.0152</td>
<td><strong>0.0046</strong></td>
</tr>
<tr>
<td><strong>EMODS–N_A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.1586</td>
<td>0.3879</td>
<td>0.4450</td>
<td><strong>0.5568</strong></td>
<td>0.4168</td>
<td>0.3554</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.009</td>
<td><strong>0.015</strong></td>
<td>0.0172</td>
<td>0.0063</td>
<td>0.0102</td>
<td>0.0119</td>
</tr>
<tr>
<td><strong>N–CMA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.296</strong></td>
<td>0.247</td>
<td>0.412</td>
<td>0.484</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.012</td>
<td>0.061</td>
<td>0.022</td>
<td><strong>0.007</strong></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td><strong>DIOP</strong> (w_o = 0.91)</td>
<td>Mean</td>
<td>*</td>
<td><strong>0.66</strong></td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>*</td>
<td>0.06</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 4.8: Average Minimum (AM) Results (** indicates the best score on that test problem)

<table>
<thead>
<tr>
<th></th>
<th>Two–on–One</th>
<th>Omni–Test</th>
<th>Lamé</th>
<th>EBN</th>
<th>LaméScaled</th>
<th>DTLZ2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMODS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.131</td>
<td>1.663</td>
<td>0.632</td>
<td>0.706</td>
<td>0.544</td>
<td>0.426</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.008</td>
<td>0.085</td>
<td><strong>0.037</strong></td>
<td><strong>0.007</strong></td>
<td><strong>0.015</strong></td>
<td>0.019</td>
</tr>
<tr>
<td><strong>EMODS–N_A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.048</td>
<td>1.279</td>
<td>0.455</td>
<td>0.825</td>
<td>0.433</td>
<td>0.115</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td><strong>0.001</strong></td>
<td>0.203</td>
<td>0.064</td>
<td>0.032</td>
<td>0.036</td>
<td><strong>0.008</strong></td>
</tr>
<tr>
<td><strong>EMODS–NN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.162</strong></td>
<td><strong>2.124</strong></td>
<td><strong>0.705</strong></td>
<td><strong>0.921</strong></td>
<td><strong>0.657</strong></td>
<td><strong>0.382</strong></td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.011</td>
<td><strong>0.0778</strong></td>
<td>0.049</td>
<td>0.034</td>
<td>0.033</td>
<td>0.028</td>
</tr>
</tbody>
</table>

Table 4.7 shows the EMODS–NN obtained during a typical run of EMODS on this problem. EMODS found the nondominated front in the lower-left part of the graph and maintained solutions in the local front, in the upper right part of the graph. Similarly, Table 4.7 shows that, except for the Two-On-One problem, EMODS is competitive with or better than N–CMA on the Shir Diversity metrics.

The results of the AM metric (Table 4.8) show that solutions tended to be fairly well separated from their neighbors in the decision space on average. An exception occurred with EMODS–N_A across all test problems, where minimum distances (MM) became small (Table 4.9).

The ADNO metric (Table 4.10) shows that on some problems, distinct alternatives are available for many of the solutions. The difference in distances between this metric and the AD metric shows
Figure 4.8: Final Nearly–Nondominated Set (EMODS–NN) for Two–On–One Problem

Table 4.9: Minimum Distance (MM) Diversity Results (bold indicates the best score on that test problem)

<table>
<thead>
<tr>
<th></th>
<th>Two–on–One</th>
<th>Omni–Test</th>
<th>Lamé</th>
<th>EBN</th>
<th>LaméScaled</th>
<th>DTLZ2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMODS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.026</td>
<td>0.026</td>
<td>0.024</td>
<td>0.065</td>
<td>0.066</td>
<td>0.059</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td><strong>0.002</strong></td>
<td><strong>0.007</strong></td>
<td><strong>0.007</strong></td>
<td>0.033</td>
<td>0.019</td>
<td>0.009</td>
</tr>
<tr>
<td><strong>EMODS–NA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.026</td>
<td>0.027</td>
<td>0.025</td>
<td>0.158</td>
<td>0.066</td>
<td>0.059</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td><strong>0.002</strong></td>
<td>0.009</td>
<td><strong>0.007</strong></td>
<td>0.062</td>
<td><strong>0.019</strong></td>
<td><strong>0.008</strong></td>
</tr>
<tr>
<td><strong>EMODS–NN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.028</strong></td>
<td>1.535</td>
<td><strong>0.041</strong></td>
<td>0.380</td>
<td>0.184</td>
<td>0.077</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.003</td>
<td>0.026</td>
<td>0.029</td>
<td>0.184</td>
<td>0.092</td>
<td>0.017</td>
</tr>
</tbody>
</table>
Table 4.10: Average Distance from Nearest Objective (ADNO) Diversity Results (bold indicates the best score on that test problem)

<table>
<thead>
<tr>
<th></th>
<th>Two–on–One</th>
<th>Omni–Test</th>
<th>Lamé</th>
<th>EBN</th>
<th>LaméScaled</th>
<th>DTLZ2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMODS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.685</td>
<td><strong>4.759</strong></td>
<td>3.099</td>
<td>1.645</td>
<td>1.134</td>
<td><strong>0.872</strong></td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.089</td>
<td>0.253</td>
<td>0.176</td>
<td>0.019</td>
<td>0.045</td>
<td>0.052</td>
</tr>
<tr>
<td><strong>EMODS–N_A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>0.051</td>
<td>3.759</td>
<td>2.609</td>
<td>1.319</td>
<td>0.918</td>
<td>0.131</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td><strong>0.002</strong></td>
<td>0.526</td>
<td>0.222</td>
<td><strong>0.062</strong></td>
<td>0.082</td>
<td></td>
</tr>
<tr>
<td><strong>EMODS–NN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td><strong>0.846</strong></td>
<td>4.679</td>
<td>3.024</td>
<td>1.449</td>
<td>1.109</td>
<td>0.549</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.132</td>
<td>0.312</td>
<td>0.213</td>
<td>0.055</td>
<td>0.057</td>
<td>0.044</td>
</tr>
</tbody>
</table>

that generally good quality alternative solutions are available in the final solution sets obtained by EMODS. For instance, for the EBN problem, Table 4.8 shows that the AD value in the archive set (0.706) is less than half the value for ADNO (1.645, Table 4.10). This provides some indication that the decision space neighbors of a solution were different than objective space neighbors of that solution. Thus, substitutable solutions could provide strong alternatives.

4.7 Observations

Identifying alternative solutions by promoting decision space diversity in an EMO algorithm is a challenging, but worthwhile, task due to the benefits provided when addressing real–world problems. While past efforts have sought to increase the overall spread of decisions, a set of solutions that is well–spread in the decision space does not necessarily contain substitutable, alternative solutions. This paper has proposed the ADNO metric that measures the distinctness of alternative solutions, as well as the AD and MM metrics to aid in quantifying the spread of decisions. These metrics consider different the aspects of solution diversity for multiobjective optimization problems that cannot be fully captured by a single metric. For instance, the use of multiple metrics in the numerical experiments conducted in this paper on the Omni–Test problem show that pairwise diversity metrics in the decision space can inaccurately characterize the diversity of a final solution set.

Experimental results also show that EMODS performs competitively with other diversity enhanc-
ing EMO algorithms for both Pareto optimality and decision space diversity. The algorithm generally achieves equivalent or better values on the metrics that quantify the spread of solutions in the decision space (i.e., Solow–Polasky diversity and Shir Diversity). This is true for all three of the final solution sets selected from the archive of EMODS. A comparison of the performance of EMODS on all problems (except Two–On–One) on the the ADNO and AD metrics shows that ADNO is always larger than AD for any final solution set. This advantage in the ADNO metric provides evidence that the custom operators in the algorithm produce solutions that are distinctly different in the decision space from solutions for which they may substitute, i.e., truly alternative solutions.

There are issues that remain to be addressed in EMO algorithms for generating alternative and diverse solutions and in the design of EMODS specifically. There are a number of diversity metrics for analyzing different aspects of solution diversity, and a comprehensive analysis of these metrics may lead to the development of better metrics for representing different aspects of decision space diversity. In EMODS, the relaxation parameter for determining which solutions are nearly–nondominated, while useful in the algorithm, might be non–intuitive for an end–user. Additionally, the crowding distance, used as an estimate of the decision space diversity when selecting parents, has been found to have issues as the number of variables increases (Deb and Tiwari, 2008), and the effectiveness of alternate diversity estimates should be compared. Finally, EMODS is an algorithm whose design is motivated by necessities in real–world multiobjective problems. As such, EMODS should be tested on problems representative of real–world engineering challenges to better explore the effectiveness of the algorithm. Ongoing work is applying EMODS to the study of a wastewater system design problem and a portfolio selection problem.
5.1 Introduction

Resilience is a multi–faceted concept. As such, measuring the resilience for a system requires multiple quantitative measures. Previous modeling of pre–hazard improvements to the civil infrastructure system (CIS) has not addressed explicit multiobjective optimization of different performance measures, or metrics, of resilience.

The resilience of a system may be measured over a single event or multiple events (Piper et al., 2013a,b). Preparing for multiple events allows planners to achieve long–term resilience through incremental, stage–wise improvements. The disadvantage to incremental improvements is that the short–term resilience of the system may receive less focus. An optimization model considering multiple aspects of resilience should include both long–term and short–term system performance.

Unfortunately, solving the true problem requires more than obtaining a solution to a multiobjective optimization model. Improvements to the CIS must often be made to publicly owned systems. These systems have multiple stakeholders that may have competing or contradictory criteria for the
necessary lifeline services that the CIS provides. Inclusion of all objectives may make the solution of the decision model computationally intractable. Furthermore, certain criteria may not be easily quantifiable, such as social or political will. Thus, a model must necessarily be an incomplete representation of the problem and leave aspects of the true problem unmodeled.

Models may be used to generate alternative solutions to provide decision makers the ability to explore the decision space and locate solutions that may better address unmodeled traits. In this paper, the evolutionary multiobjective optimization (EMO) algorithm EMODS (EMO for Diverse Solutions) described in Piper and Ranjithan (2013) (Chapter 4) is applied to an example CIS to find solutions that improve system-wide resilience according to multiple metrics. The example CIS, Coastal City, was first presented in Piper et al. (2013a) (Chapter 2) and Piper et al. (2013b) (Chapter 3). Short- and long-term resilience of the system is measured by these objectives. The use of EMODS makes possible the simultaneous discovery of a front of nondominated solutions and location of alternative solutions that are distinct and diverse from other solutions. The trade-offs among the solutions, both nondominated and nearly-nondominated, are analyzed and compared to demonstrate the value obtained from pursuing solution diversity.

First, a review of the past studies is given on CIS, EMO, and pursuing solution diversity. Then, the CIS modeling framework introduced in Piper et al. (2013a) is presented and applied to the Coastal City CIS. EMODS is described and adapted for the discrete decision variables used in the CIS resilience improvement problem. Numerical experiments are conducted for two and four objective problems on the Coastal City CIS. The effectiveness of EMODS is evaluated and compared to the well-known NSGA-II genetic algorithm for multiobjective problems in terms of both Pareto front convergence and solution diversity. Finally, the performance of improvement plans obtained on the two-objective experimental set are evaluated on a three-objective problem instance with alternative hazard scenarios. The results are analyzed to determine what effect diversity has on improving the robustness of the solution set that the EMO algorithms obtain.
5.2 Background

5.2.1 CIS Resilience

There is little consensus in literature about how to quantify resilience of the CIS. Part of this may be due to the vast number of definitions of resilience that exist. Zhou et al. (2010) compiled a list of twenty-seven qualitative and quantitative definitions of resilience appearing in academic literature in various fields. Many definitions are related, but it is likely that a model for improving CIS resilience would have unmodeled objectives when each stakeholder may hold a different view of what truly constitutes resilience in a system.

Pendall et al. (2009) also reviewed definitions and concepts of resilience in different fields. The study argues that resilience could provide an example of a “fuzzy concept” defined by Markusen (1999). Fuzzy concepts are ill-defined and therefore difficult to effectively quantify. Cutter et al. (2010) indirectly demonstrated the difficulty of quantifying resilience through the development of a disaster resilience index. In Cutter et al. (2010), seven infrastructure factors influence the resilience of a community (along with many other non-infrastructure related factors). Factors are chosen based on availability of data and the existence of literature arguing for each factor’s importance. There is inherent subjectivity with such a method for quantifying resilience. These past efforts show that models for improving resilience in CIS must necessarily have unmodeled aspects.

Few studies have explicitly considered optimizing simultaneously for the different aspects of resilience. Most CIS optimization models measure a single aspect of resilience or performance (e.g., Cavdaroglu et al. 2011; Miller-Hooks et al. 2012; Peeta et al. 2010; Xu et al. 2007). Despite this, resilience is generally considered multi-faceted (e.g., Bruneau et al. 2003; Rinaldi et al. 2001). Relationships and conflicts between resilience metrics have not been analyzed, to the authors’ knowledge. The management of CIS has been addressed with EMO. In Okasha and Frangopol (2009), a genetic algorithm is used to select designs and maintenance routines with criteria of life-cycle cost, redundancy, and system reliability; however, the application is limited to a detailed view of a single component of the CIS.
The improvement of CIS resilience, on the other hand, has focused mainly on decisions to optimizing post–hazard recovery actions (e.g., Cavdaroglu et al. 2011; Lee et al. 2007; Miller-Hooks et al. 2012). Efforts with an emphasis on pre–hazard improvements tend to consider a limited subsystem of the larger CIS, like bridges (e.g., Fan and Liu 2008; Fan et al. 2010; Karlaftis et al. 2007). The CIS modeling framework presented in Piper et al. (2013a) and Piper et al. (2013b), on which the optimization problems in this paper are based, has mechanisms for representing multiple infrastructure systems within the overall CIS. Additionally, the modeling framework is capable of explicitly representing the interdependencies between systems and taking advantage of these interdependencies to modify component, and thus system, performance.

### 5.2.2 Alternative Solutions for Incompletely Modeled Multiobjective Problems

Generating alternative solutions is viewed as an acceptable strategy for handling incompletely modeled problems. The method of modeling to generate alternatives (MGA) was presented by Brill (1979). MGA uses mathematical programming to find solutions within a given relaxation of the optimal solution, but maximally different in the decision space. These resulting solutions are alternatives that may better meet aspects of the problem not captured by the model. For instance, Abdulkadri and Ajibefun (1998) uses MGA for farm management, and the alternative solutions provide farmers an array of choices achieving good levels of profit but had different levels of crop diversification to reduce risk to the farmers. A sample of studies applying MGA includes airline routing (Brill et al., 1990), wastewater management (Zechman and Ranjithan, 2004), air quality management (Loughlin et al., 2001), and solid waste management (Kaplan and Ranji Ranjithan, 2007; Kaplan et al., 2009; Yeomans, 2011).

Generation of alternative solutions to improve the handling of unmodeled problem characteristics has been a recent topic in multiobjective evolutionary algorithms. Generally, these approaches attempt to both approximate the Pareto front and increase the diversity of the solutions. Different approaches have been reported (e.g., Kramer and Danielsiek 2010; Shir et al. 2009; Ulrich et al. 2010a,b; Ursem and Justesen 2012). All the cited approaches only apply to continuous variable optimization.
5.3 Civil Infrastructure System Modeling Framework

The CIS modeling framework used in this paper was originally introduced in Piper et al. (2013a). The framework consists of six major concepts, shown in Figure 5.1. Each concept may have different implementations and realizations depending on the critical aspects of resilience examined and the configurations of the CIS.

In this paper, the CIS of a community called Coastal City, shown in Figure 5.2, will be modeled. The components of Coastal City are based on a real coastal community. Coastal City’s CIS consists of a levee and a road system, where the levee protects certain road components from flooding.

The modeling of Coastal City with the framework from Piper et al. (2013a) is described in rest of this section. The notations used in this section are summarized in Table 5.1.

5.3.1 Hazard Scenarios

In the problem instances addressed in this paper, the resilience of Coastal City will be optimized for $\epsilon$ total hazard events. The index $e = 1, \ldots, \epsilon$ indicates how many events have occurred. Each hazard
event may occur as one of multiple possible hazard scenarios contained in the set $H^e$ for $e \geq 1$. Each hazard scenario $h \in H^e$ has a probability of occurring in its particular event, $\pi^e_h$. Each hazard scenario has a deterministic effect on the status of CIS components, which would need to be calculated beforehand through simulation or gathered from historic hazard data.

### 5.3.2 System Representation

Let $C$ be the set of all components of the CIS, and let $L$ and $R$ be the sets of levee and road components, respectively, so that $C = L \cup R$ in Coastal City. For any component $c \in C$, the service level of a component is the amount of a particular type of service that the component provides. For a levee component, service level indicates how much protection (from 0% to 100%) the levee provides the road components it protects in a hazard scenario. For road components, service level is the number of vehicles/hour that a road segment can support. The set $K$ contains all types of services in the Coastal City CIS. The service level for usage type $k \in K$ of component $c$ in hazard scenario $h$ and event $h$ is denoted as $s^e_{h,c,k}$. The maximum (or base) service level of a component is given as $\sigma_{h,c}$.

The relationships among components need to be specified to account for component interde-
Table 5.1: CIS Modeling Framework Notation for Multi–Event Resilience Metrics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>components of CIS, indexed by $c$</td>
</tr>
<tr>
<td>$e$</td>
<td>events occurred, $e = 0, 1, \ldots, \epsilon$</td>
</tr>
<tr>
<td>$H^e$</td>
<td>hazard scenarios possible for event $e$, indexed by $h^e$</td>
</tr>
<tr>
<td>$K$</td>
<td>usage type, indexed by $k$</td>
</tr>
<tr>
<td>$\pi_h^e$</td>
<td>probability of hazard scenario $h$ in event $e$</td>
</tr>
<tr>
<td>$TU_c$</td>
<td>set of components whose services can physically reach component $c$</td>
</tr>
<tr>
<td>$TD_c$</td>
<td>set of components physically reachable by services from component $c$</td>
</tr>
<tr>
<td>$SU_c$</td>
<td>set of components with functional linkages going to component $c$</td>
</tr>
<tr>
<td>$SD_c$</td>
<td>set of components with functional linkages starting from component $c$</td>
</tr>
<tr>
<td>$\sigma_{c,k}$</td>
<td>base (maximum) service level of usage $k$ provided by component $c$</td>
</tr>
<tr>
<td>$d_{h,c,k}^e$</td>
<td>service type $k$ supplied or demanded at component $c$ in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$f_{h,c,c',k}^e$</td>
<td>potential usage flowing from component $c$ to component $c'$ for service $k$ in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$A_h$</td>
<td>set of components losing performance directly from hazard in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$B_h$</td>
<td>set of components losing performance from cascading failures and dependencies in scenario $h$ after event $e$</td>
</tr>
<tr>
<td>$\phi_{h,c,k}^e$</td>
<td>initial service level of usage $k$ at component $c$ in scenario $h$ (when $c \in A_h$) after event $e$</td>
</tr>
<tr>
<td>$s_{h,c,k}^e$</td>
<td>service level achievable for usage $k$ at $c$ in scenario $h$ after improvements after event $e$</td>
</tr>
</tbody>
</table>

Dependencies, potential service usage, and movement of the populace. In the modeling framework, components are connected by directed links. Links are divided into two types: transport links and service links. The transport links in Coastal City represent which road segments are connected, and where vehicles can travel. The service links in the Coastal City CIS represent which levee segments protect which road segments.

Formally, sets $TD_c$, $TU_c$, $SD_c$, and $SU_c$ are defined for components. For $c \in R$, the sets $TD_c$ and $TU_c$ consist of all $c' \in R$ such that $c'$ is directly connected to road segment $c$. Also for $c \in R$, $SU_c$ consists of levee components $c'$ such that $c'$ protects $c$ from flooding. For $c \in L$, $SD_c$ consists of $c' \in R$ such that $c'$ is protected by $c$.

Service usage types, $K$, are divided into two categories. The first category, $K_S = \{k_0, k_1\}$, contains the vehicles/hour capacity of road segments, $k_0$, and the protection that the levees give the road segments.
system, $k_1$. The second category, $K_T$, consists of the trips that the population would take on the road network, which is divided further into fifteen types of trips. A summary of the types of trips is omitted. Example trips might include commuting to work or school, daily errands, shipping goods, etc.

### 5.3.3 Component Status Calculation

The original framework introduced a general notion of determining component status based on the effects of a hazard scenario and interdependencies. A simplified instantiation of that general notion is presented here.

When a component $c$ fails in hazard scenario $h \in H^e$, the service level of $c$, $s_{h,c,k}^e$, is set to 0. This indicates that without any improvements, components that lose service in hazard scenario $h$ completely fail, which is all components $c \in A^e_h$. For components that are not directly affected by the hazard scenario, $c \in B^e_h$, service level is determined by interdependencies. The effects of interdependencies on service level of a road segment, given as $s_{h,c,k}^e(SU_c)$, are represented by the following equation:

$$s_{h,c,k_0}^e(SU_c) = \sigma_{c,k_0} \min_{c' \in SU_c} \left\{ \frac{s_{h,c',k_1}^e}{\sigma_{c',k_1}} \right\}, \quad \forall c \in R, e = 1, \ldots, \epsilon, h \in H^e$$

where $c$ is a particular component in $R$. Section 5.3.5 demonstrates how the effects of improvement choices are included for components in $A^e_h$ and $B^e_h$.

### 5.3.4 CIS Resilience Metrics

Numeric experiments in Section 5.5 will consider up to four metrics: three resilience metrics and the cost of improvements. The metrics are defined and explained in Piper et al. (2013a) and Piper et al. (2013b). Brief explanations and definitions of the chosen resilience metrics are provided here for the Coastal City CIS.
The weighted serviceability of the road system is defined by the following equation:

\[
S_{CIS,W} = \frac{1}{\varepsilon} \sum_{e=1}^{E} \sum_{h \in H_e} \pi^e_h \frac{1}{|R|} \sum_{c \in R} s^e_{h,c,k}.
\]  

(5.2)

which calculates the expected available service level of the road system. The serviceability metric does not measure whether the population is able to use the road system to take the trips they desire. To quantify how well the population can complete desired trips, the demand satisfaction metric is defined. Demand satisfaction for a particular hazard scenario is determined by solving the following multicommodity network flow problem:

\[
\text{minimize } D(e, h) = \sum_{k \in K_T} \sum_{c \in C} \left( d^e_{h,c,k} - \sum_{c' \in T_U_c} f^e_{h,c',c,k} \right)
\]

(5.3a)

subject to

\[
\sum_{c' \in T_D_c} f^e_{h,c',c,k} - \sum_{c' \in T_U_c} f^e_{h,c',c,k} \geq d^e_{h,c,k} \quad \forall c \in R, k \in K_T
\]

(5.3b)

\[
\sum_{k \in K_T} \sum_{c' \in T_D_c} f^e_{h,c',c,k} \leq s^e_{h,c,k_0} \quad \forall c \in R
\]

(5.3c)

\[
f^e_{h,c',c,k} \geq 0 \quad \forall c \in C, c' \in T_D_c, k \in K_T
\]

(5.3d)

where \(D(e, h)\) is the demand unsatisfied in event \(e\) if hazard scenario \(h\) occurs. The number of trips starting or ending at road segment \(c\) in event \(e\) and hazard scenario \(h\) is given by \(d^e_{h,c,k}\) where \(k \in K_T\). If trips of type \(k\) start at \(c\), then \(d_{h,c,k} < 0\); whereas if trips of type \(k\) end at \(c\), then \(d_{h,c,k} > 0\). Equation (5.3b) maintains flow conservation as the vehicles move around the network. Equation (5.3c) constrains the maximum number of vehicles/hour leaving a road segment to the actual capacity of that road segment. The values of \(s^e_{h,c,k_0}\) in that equation are determined for each hazard scenario by Equation (5.1).

It is important to note that this metric does not attempt to find an optimal arrangement of vehicle traffic. It only attempts to determine if there is sufficient capacity to accommodate particular trips desired by the population. Motivating this distinction is the recognition that modeling human behavior is difficult, perhaps especially in the aftermath of a natural disaster, and outside the scope
of this paper. As a result, this paper is only concerned with determining how many trips a feasible
usage pattern can satisfy.

The final resilience metric that will be used is serviceability recovery time, which measures how
long it takes the road system to return to full vehicle capacity. If $R(e, h)$ is the recovery time for hazard
scenario $h \in H^e$, the multi–event recovery time is:

$$
\frac{1}{\varepsilon} \sum_{e=1}^{\varepsilon} \sum_{h \in H^e} \pi^e_h R(e, h).
$$

(5.4)

There is no general method for calculating recovery time. Recovery time calculation will depend on
the system, the system’s resources, and what service level is considered sufficiently recovered. The
implementation of recovery time used for this paper is a dynamic, state–saving, breadth–first graph
traversal algorithm that simulates the movement of recovery crews across the CIS network and their
recovery process. Two components of the road network, on the far northwest and northeast parts
of the Coastal City region, are designated as recovery bases from where the recovery crews begin
restoration in the event that a hazard occurs. Each base has 100 crews, each capable of restoring a
single reachable and failed component every time period. From each recovery base a random set of
failed and reachable components (less than the number of crews) were restored in each time period.
When bases have overlapping reachable road components, bases are combined, and crew resources
are pooled.

5.3.5 Improvement Decisions

For each component, levee and road, it is assumed that a certain set of improvement choices can
be applied that may improve the worst service level that the component provides after a natural
hazard. These improvement decisions may include raising or rebuilding sections of the levee, raising
road beds, building better drainage or clearing materials, like trees, from the side of roads that could
block travel after a hazard.

For component $c$, $U^e_c$ is the set of improvements that can be made in stage $e$, $e = 0,\ldots, \varepsilon - 1$. 

119
Each improvement choice is considered a different level of decision that could be made. Level 0 is always considered the “do–nothing” decision. Define $x_{e,c,u}$ to be a binary decision variable that is 1 if improvement $u$ is chosen in stage $e$ and 0 if not. Let $\lambda_{h,c,u,k}^e$ be the effect on service usage type $k$ for component $c$ in scenario $h$ and event $e$. Note that $\lambda_{h,c,u,k}^e$ could be equal to 0 for certain components and hazard scenarios, because certain improvements may not be effective depending on the hazard scenario. It is assumed that if a component is improved, then the effect of the improvement in event $e$ and scenario $h$ is the same regardless of what stage the improvement was chosen.

Though the set of possible upgrades, $U_{e,c}^e$, could potentially change as new technologies become available, these sets were fixed for all $e$ in the numerical experiments described in Section 5.5. The cost of each improvement, $\kappa_{e,c,u}^e$, used for calculating the improvement cost objective function, is similarly fixed for each $e$, as it was assumed that technologies did not change during the time period in the study. Furthermore, it is assumed that each component can receive at most one improvement.

For components that fail in hazard scenario $h \in H^e$, $c \in A_{h,c}^e$, the effect of improvements is given by

$$s_{h,c,k}^e = \sum_{e'=0}^{\epsilon-1} \sum_{u \in U_{e,c}^e} x_{e',c,u}^e \lambda_{h,c,u,k}^e \quad \forall c \in C, e=1,\ldots,\epsilon, h \in H^e$$

(5.5)

Components that do not fail in a hazard scenario, $c \in B_{h,c}^e$, have a slightly more complex determination of service level after improvement effects are considered:

$$s_{h,c,k_0}^e(SU_c) = \sigma_{c,k_0} \min_{c' \in C} \left\{ \frac{s_{h,c',k_0}^e s_{h,c',k_0}^e}{\sigma_{c',k_1}} \right\} + \sum_{e'=0}^{\epsilon-1} \sum_{u \in U_{e,c}^e} x_{e',c,u}^e \lambda_{h,c,u,k}^e \quad \forall c \in R, \forall c' \in SU_c, e=1,\ldots,\epsilon, h \in H^e$$

(5.6)

With complex cascading failures, the process of determining service level due to interdependencies while including improvement effects may need to be more complex. In the Coastal City CIS, however, there is only one level of cascading failures that could occur: from levee components to road components. This lessens the complexity needed to determine service level of road components.
5.4 EMODS Description

This section describes the Evolutionary Multiobjective Optimization Algorithm for Diverse Solutions, EMODS, for optimizing CIS resilience. General evolutionary algorithms are population-based metaheuristics that operate on principles similar to Darwinian natural selection, such that the fittest solutions survive and reproduce over successive generations. The outline for EMODS is shown in Figure 5.3, and it shares many of the same characteristics with general evolutionary algorithms. An evolutionary algorithm is composed of various operators, each of which has a specific purpose in the search for nondominated solutions. Operators include selection (determines the solutions that have better ranking for breeding), crossover (analogous to reproduction, creates new solutions from parent solutions), and mutation (randomly changes solutions in ways that may or may not be beneficial). Additional operators, called archiving or elitism, are used to preserve good solutions, where the criteria of “good” depends on the specific algorithm and potentially the specific problem.

EMODS as described in Piper and Ranjithan (2013) is implemented for problems with continuous decision variables, and the algorithm must be adapted to handle discrete optimization problems. For instance, the archive excludes overlapping and inferior solutions from insertion based on Euclidean distances between variables (see Figure 3 in Piper and Ranjithan (2013)). Using this scheme with discrete variables could be problematic, since solutions differing only in the improvement chosen for a single component could be considered sufficiently different for both to be included in the archive.

Some preliminary notation for EMODS will be defined here. Let $P$ be the population of solutions in each generation, its size is given by the parameter $t_p$. EMODS also maintains an archive of nondominated and nearly-nondominated solutions that are maximally different in the decision space. Let $A$ be the archive, and $t_A$ be the maximum desired size of the archive while $t_N$ is the target number of nondominated solutions for the archive (so $t_D = t_A - t_N$ is the number of nearly-nondominated, diversity solutions desired). EMODS also uses a relaxation parameter $r \in [0, 1]$ which indicates the allowable relaxation for solutions to be considered nearly-nondominated. For example, $r = 0.10$ would permit solutions with fitness within a 10% relaxation of a nondominated solutions to be considered
Figure 5.3: Outline of EMODS Algorithm
nearly-nondominated.

5.4.1 Solution Encoding and Variation Operators

A potential solution, or individual, in an evolutionary algorithm is represented by an encoding of the decisions for that solution. The choice of encoding is critical to enable the variation operators (crossover and mutation) to effectively explore and exploit the solution space to locate good solutions.

A CIS improvement plan will be encoded as three vectors. This encoding can be used by evolutionary algorithms capable of handling discrete decision variables and is not specific to EMODS. If \( C \) is the set of components that can be upgraded, then let \( \mathbf{v}, \mathbf{w} \) and \( \mathbf{y} \) be row vectors of size \(|C|\). In \( \mathbf{v} \), the \( c^{th} \) entry, \( v_c \), is a binary variable that is a 1 if component \( c \) will receive an upgrade and a 0 if not. If \( v_c = 1 \), then \( y_c \) represents the level of the upgrade that component \( c \) will receive, where \( y_c \in U_c \setminus \{0\} \), the set of possible upgrades for component \( c \) without the “do–nothing” upgrade. Finally, \( w_c \) is the stage at which component \( c \) will receive the upgrade, where \( w_c \in \{0, \ldots, \epsilon - 1\} \), assuming that there are \( \epsilon \) hazard events that will occur.

In short, this encoding implies that if \( v_c = 1, y_c = u, \) and \( w_c = e \), then \( x_{e,u}^{c} = 1 \). Conversely, if \( x_{e,u}^{c} = 1 \) and \( u > 0 \) (is not the “do–nothing” decision), then \( v_c = 1, y_c = u, \) and \( w_c = e \), but if \( x_{e,u}^{c} = 0 \) or \( u = 0 \) when \( x_{e,u}^{c} = 1 \), then \( v_c = 0 \) and the values of \( y_c \) and \( w_c \) will not matter.

The correspondence between an encoding and an improvement plan is not bijective, it is surjective, because every possible improvement plan can be represented. A given plan, however, will correspond to multiple encodings if it has at least one “do–nothing” decision.

For the variation operators, uniform crossover and a bit flip mutation are used. Uniform crossover is demonstrated in Figure 5.4 for two solutions in a three–component CIS. The uniform crossover operator takes two proposed improvement plans, \( S_1 = \{\mathbf{v}^1, \mathbf{y}^1, \mathbf{w}^1\} \) and \( S_2 = \{\mathbf{v}^2, \mathbf{y}^2, \mathbf{w}^2\} \), and generates crossover template vector, \( \mathbf{t} \), of size \(|C|\). Each element of \( \mathbf{t}, t_i, i = 1, \ldots, |C| \), is randomly set equal to 1 if \( S_1 \) and \( S_2 \) should exchange the values \( v_i^1 \) and \( v_i^2 \), \( y_i^1 \) and \( y_i^2 \), and \( w_i^1 \) and \( w_i^2 \), and 0 if the solutions keep their values for component \( i \). The probability that \( t_i = 1 \), \( p_t \), is a parameter that must be set by
a user.

The bit flip mutation operates on a single solution, $S$, at a time. For each component $i$, $v_i$ is set to $1 - v_i$ with probability $p_f$. Separately, there is a probability that the improvement level $y_i$ changes to any one of the other valid improvement levels, and $w_i$ changes another stage for the improvement. Note that changing the choice to make an improvement, the level of an improvement, and the stage at which an improvement is selected are independently determined. Again, parameter $p_f$ must be set beforehand.

5.4.2 Decision Diversity Metric

As the goal of using EMODS is to find solutions diverse in the decision space, a metric to quantify distance between solutions and another to quantify diversity needs to be defined. First, a decision space distance between the solutions $S_1 = \{v^1, y^1, w^1\}$ and $S_2 = \{v^2, y^2, w^2\}$ is defined. Decision distance between these solutions is defined similarly to the Hamming distance as:

$$d(S_1, S_2) = \sum_{c \in C} |v^1_c y^1_c - v^2_c y^2_c| + |v^1_c w^1_c - v^2_c w^2_c|$$  \hspace{1cm} (5.7)

An advantage of using this metric is that it is as fast to calculate as it can be ($O(|C|)$) while considering improvements made for each component. Some possible distances calculated between two solutions for a single component are shown in Table 5.2. The table shows distances for every possible combination of improvements for a single component from two separate solutions when there are two stages (two hazard events) and three improvement options (including the do–nothing improvement).

Table 5.2 shows that the same improvement decision made in a different stage has a different effect on the distance between two solutions. For example, suppose two solutions choose upgrade level 2 ($u = 2$) for the same component. If both solutions made the improvement in stage 0, then the distance between the solutions is 0. If, instead, one solution made the improvement in stage 1,
Figure 5.4: Crossover Operator for CIS Encoding
Table 5.2: Decision Distances Between Component Improvement Options

<table>
<thead>
<tr>
<th>Stage = 0</th>
<th>Stage = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u = 0$</td>
<td>$u = 0$</td>
</tr>
<tr>
<td>$u = 1$</td>
<td>$u = 1$</td>
</tr>
<tr>
<td>$u = 2$</td>
<td>$u = 2$</td>
</tr>
</tbody>
</table>

then the distance between the solutions is 1. The largest difference possible is when one solution chooses a high-level decision in a late stage, while another solution chooses no improvement for a component. This can be considered akin to saying that planning to make an improvement in later stages is more dissimilar from choosing no improvement than making that improvement in earlier stages.

Measuring decision diversity of a set of solutions requires measuring diversity between solutions that are substitutable. A solution may be substituted for another solution (and chosen for implementation) if their objective function values are similar. For a multiobjective problem, substitutability between two solutions requires that the solutions are similar in all their objective function values. What a decision diversity metric needs to calculate is the distances between substitutable solutions.

Several diversity metrics were described by Piper and Ranjithan (2013). Only one of them considered substitutability (the Average Distance from Nearest Objective (ADNO) metric), which is mathe-
matically expressed for a population \( P \) of solutions as:

\[
ADNO(P) = \frac{1}{|P|} \sum_{S \in P} d\left(S, \arg\min_{S' \in P, S' \neq S} d(S, S')\right).
\]  

(5.8)

where \( \mathbf{f}(S) = (f_1(S), f_2(S), \ldots, f_m(S)) \) is a vector of the objective function values for solution \( S \) of the multiobjective problem. For the CIS optimization problem, the distance function \( d(\cdot, \cdot) \) is defined in Equation (5.7). It is also possible to take the decision space distance from a solution to its nearest objective space neighbor, which is written as \( adno(S) \). This single–solution version of \( ADNO \) will be used in the selection operator.

### 5.4.3 Selection Operator

EMODS uses a novel hypervolume–based fitness that ranks a population of solutions based on a scalarization of their fitness, or objective values. Hypervolume is a measurement of the size of the space dominated by a set of solutions with respect to a reference point that defines the extents of the region. Use of hypervolume in multiobjective optimization began as a performance measure of algorithms (Zitzler and Thiele, 1999), and now some EMO algorithms use hypervolume (or something inspired by it) within an evolutionary algorithm to help guide the search (Bader and Zitzler, 2011; Beume et al., 2007; Dorn, 2005). The hypervolume of a set \( F \) is denoted as \( \text{Hy}_p(F) \).

Typically, hypervolume is measured for a set of nondominated solutions. In Piper and Ranjithan (2013), the induced front hypervolume is defined for the induced front derived from any solution in a population.

**Definition (Induced Front).** If \( F \) is the set of nondominated solutions in a population of solutions, \( P \), let \( D_x = \{ p \in F \mid p \preceq x \} \) be the solutions in \( F \) that dominate \( x \in P \). Then the set \( F_x = (F \setminus D_x) \cup \{x\} \) is defined as the induced front of \( x \).

The induced front for a solution \( S \) is formed if the solutions dominating \( S \) are temporarily removed from the nondominated front, and \( S \) is added to the resulting set. EMODS calculates the hy-
Figure 5.5: Induced front hypervolume of solution $a$. Area of shaded region is induced front hypervolume of solution $a$. Both $f_1$ and $f_2$ are to be minimized.

The induced front hypervolume of a solution is used a fitness measure to select parent solutions for crossover and mutation. Selection is accomplished through a binary tournament with two sets of selection criteria. The first set of selection criteria applies until $\lfloor t_p/2 \rfloor$ parents are selected; the criteria in order are: induced front hypervolume and exclusive hypervolume contribution (nondominated solutions only). Exclusive hypervolume contribution only applies to nondominated solutions. This quantity compares two nondominated solutions by calculating how much of the hypervolume of the nondominated front is exclusively dominated by each solution. The solution with the higher exclusive hypervolume contribution is chosen as a parent. For this first set, no relaxation of induced front hypervolume is used.

The remaining $\lceil t_p/2 \rceil$ parents are selected according to the following criteria in order: induced front hypervolume and decision space diversity from $adno$. For the second set of criteria, given two
solutions, $S_1$ and $S_2$, the relaxation parameter $r$ chooses solution $S_1$ to be a parent solution if

$$Hyp(F_{S_1}) < r \cdot Hyp(F_{S_2}).$$  \hspace{1cm} (5.9)

If, however,

$$r \cdot Hyp(F_{S_1}) \leq Hyp(F_{S_2}) \leq Hyp(F_{S_1})$$  \hspace{1cm} (5.10)

then decision diversity is used to pick the winner.

Decision diversity is calculated by determining $adno(S)$ for each $S \in P$. If the relaxation parameter means that neither solution appears significantly better by induced front hypervolume, then the winning solution is chosen by the solution with the highest $adno$.

The selection operator, with both sets of criteria, picks a number of solutions equal to $t_P$, the initial size of the population, to be parents, and then applies the crossover and mutation operators to the chosen solutions to generate a new population for the next generation.

### 5.4.4 Archiving Procedure

EMODS uses an archive that is distinct from most other evolutionary algorithms in that the archive contains a set of nondominated solutions, $N_A$, and a set of diversity solutions $D_A$ such that $A = N_A \cup D_A$ and $N_A \cap D_A = \emptyset$ at each generation. The archive update procedure takes place in two major steps. The first step updates the nondominated solutions in $N_A$. The second step uses a heuristic method to select a subset of solutions from the combined archive and population of the current generation that are not only nearly-nondominated but also maximally spread through the decision space.

The archiving procedure for EMODS has been modified in two ways from its original presentation in Piper and Ranjithan (2013). First, nearly-optimal solutions are identified not via relaxation of induced front hypervolume, but by checking whether a solution is within a relaxation of every objective function value from a nondominated solution. Second, the mechanism for determining
inclusion of diversity solutions into the archive has been modified. In part, this latter change was necessary to accommodate discrete decision variables. Additionally, this change helps identify a set of solutions that are furthest apart possible given a decision distance metric like that defined in Equation (5.7).

The set $N_A$ has its desired size given by the parameter $t_N$. Initially all nondominated solutions of $P \cup A$ are selected for $N_A$. If $|N_A| > t_N$, then a trimming procedure repeatedly removes the solution from $N_A$ with the lowest exclusive hypervolume contribution to $Hy p(N_A)$.

To choose the diversity solutions, $D_A$, the merged population, $P \cup A$ must be filtered so that only nearly–nondominated solutions are considered. Given a solution $S_1 \in P \cup A$, $S_1$ is considered nearly–nondominated if there exists $S_2 \in N_A$ such that

$$f_i(S_2)(1 + r) < f_i(S_1) \quad \forall \ i = 1, \ldots, m$$

(5.11)

where $f_i(S_1)$ are objective function values, and it assumed that all objective functions are minimizations. All solutions that are nearly–nondominated are considered potential diversity solutions and put into the set $PD$. Potential diversity set $PD$ is selected so that $PD \cap N_A = \emptyset$, and if any nondominated solutions are not in $N_A$, they may be included in $PD$.

Distances between all solutions in $PD \cup N_A$ are calculated according to Equation (5.7). All potential diversity solutions are included in $D_A$ if $|PD| \leq t_D$. Otherwise, if $|PD| \geq t_A$, then a heuristic method is used to try to select a maximally diverse subset of $PD$ so that the solutions in $D_A$ are as diverse as possible from one another. Selecting a subset of no more than $t_D$ points out of $|PD|$ points such that the minimum distance between any pair of points is maximized is known as the $p$–dispersion problem. The problem is known to be NP–hard. As solving an NP–hard problem with every archive update is likely to be prohibitively expensive, especially if population sizes increase, heuristic methods described in Erkut et al. (1994) and Drosou and Pitoura (2009) are employed, namely, greedy construction combined with best pairwise interchange.

The adaptation of those heuristics for EMODS archiving is described here. Let $d_{S_1, S_2}$ be the dis-
1: $D_A = \emptyset, A = N_A$
2: \textbf{while} $|A| < t_A$ \textbf{do}
3: \hspace{1em} Find $S_1 \in PD$ s.t. $d(S_1, A) = \max\{d(S_1, S_2) \mid S_2 \in A\}$
4: \hspace{1em} $D_A = D_A \cup \{S_1\}, A = N_A \cup D_A$
5: \hspace{1em} $PD = PD \setminus \{S_1\}$
6: \textbf{end while}
7: \textbf{return} $A$

Figure 5.6: Greedy Construction Heuristic

tance between solutions $S_1$ and $S_2$. The greedy heuristic is described in Figure 5.6. As in Erkut et al. (1994), define $d(S_1, A) = \min\{d(S_1, S_2) \mid S_2 \in A, S_1 \neq S_2\}$. The greedy construction heuristic iteratively adds to the archive the solution in $PD$ that is furthest from the current archive solutions. The difference between EMODS adaptation and a standard greedy heuristic is that part of the set being selected, nondominated solutions $N_A$, is chosen before the procedure begins.

Erkut et al. (1994) suggested augmenting the greedy construction heuristic with an interchange algorithm, which, given a set, tries to improve the minimum distance between points of the set by determining how the minimum distance would change if one of the solutions closest to another in the set was switched with a solution outside of the set. The best pairwise interchange heuristic looks at all points that were not selected by the greed construction heuristic, and picks the one that improves the minimum distance by the most. The adaptation of the heuristic for EMODS is provided in Figure 5.7, where if $A$ is any set, $M(A) = \min\{d(S_j, S_k) \mid S_j, S_k \in A\}$.

The best interchange heuristic described in Figure 5.7 is more complex than the version described by Erkut et al. (1994) because it must not switch out any solutions in $N_A$. Only solutions in $D_A$ can be changed with other solutions in $D_A$, and this slightly modified heuristic method respects that requirement.

5.5 Numerical Experiments

Three groups of numerical experiments were conducted. Experimental Group 1 considered CIS resilience with two objective functions: cost and serviceability. Experimental Group 2 considered re-
1: $A = N_A \cup D_A$
2: while $A$ changes during the loop do
3:   Find $S_1, S_2$ such that $d(S_1, S_2) = \min \{d(S_j, S_k) \mid (S_j, S_k \in A) \land (S_j \notin N_A \lor S_k \notin N_A)\}$
4:   for $S' \in PD(S' \notin A)$ do
5:     if $S_1 \notin N_A$ then
6:       $D_A' = D_A \cup \{S'\} \setminus S_1$
7:     end if
8:     if $S_2 \notin N_A$ then
9:       $D_A'' = D_A \cup \{S'\} \setminus S_2$
10:    end if
11:   if $S_1 \notin N_A$ and $S_2 \notin N_A$ then
12:     if $M(D_A') > M(A)$ and $M(D_A') \geq M(D_A'')$ then
13:       $D_A = D_A'$
14:     else if $M(D_A'') > M(D_A)$ and $M(D_A'') > M(D_A')$ then
15:       $D_A = D_A''$
16:     end if
17:   else if $S_1 \notin N_A$ and $M(D_A') > M(D_A)$ then
18:     $D_A = D_A'$
19:   else if $S_2 \notin N_A$ and $M(D_A'') > M(D_A)$ then
20:     $D_A = D_A''$
21:   end if
22: end for
23: end while
24: return $A$

Figure 5.7: Best Interchange Heuristic
silence by examining four objective functions: improvement cost, serviceability, serviceability recovery time, and demand satisfaction. The third group considered the robustness of improvement plans and evaluated the contribution of diversity to finding robust final solution sets. The section concludes with a brief examination of the decision space diversity of selected solutions from the final archives.

There are two goals for optimizing CIS improvements in the Coastal City example. The first is simply to find the Pareto front (or, more likely, a good approximation of it) for the chosen objective functions. The second goal is to find diverse and substitutable solutions due to the uncertainties of future hazards.

The first goal was tested for the first two experimental groups by calculating the hypervolume of a solution set of fifty solutions obtained by EMODS and comparing it to the hypervolume obtained by two other methods. The commonly–used NSGA–II algorithm (Deb et al., 2002) was used to compare EMODS performance to other evolutionary algorithms. A random search (with a hypervolume–trimmed nondominated archive) was used to determine if the CIS improvement problem is solved well by evolutionary algorithms or if evolutionary algorithms do no better than randomly generated solutions. The computing time required for each algorithm was reported, but function evaluation time was not reported separately by method, as each method has the same number of function evaluations.

The ability of EMODS to achieve diverse solutions was evaluated by calculating the three diversity metrics for the final archives obtained by each method. The first is the $ADNO$ metric that was previously introduced. The second metric is the Average Minimum Decision Space Distance ($AM$), which was introduced by Piper and Ranjithan (2013). The third metric is the Average Decision Distance from Nondominated Neighbor ($ADNN$) metric, and it measured the average decision distance from each archive solution to the nondominated solution that is nearest in the objective space. The new $ADNN$ metric measured the diversity for solutions that are substitutable for nondominated solutions. This is distinct from the $ADNO$ metric, which measures decision differences between each solution and its most directly substitutable solution, whether or not that solution is nondominated.
Finally, the third group of experiments evaluated how the final nondominated fronts obtained by the EMO methods change for an alternative hazard scenario set. The number of solutions in the nondominated fronts under both hazard scenarios were calculated, as well as how the average number of nearly-nondominated solutions that became nondominated in when evaluated for the alternative hazard scenarios. Possible connections between the diversity of the nearly-nondominated solutions and the robustness of solutions were explored.

5.5.1 Parameters and Setup

The parameters for the Coastal City problem are summarized in Table 5.3. Each method has five randomly seeded of 500 generations with a population size of 100 (giving 50,000 function evaluations to each run).

Identical encodings were used for EMODS and NSGA–II to allow both to use the same crossover and mutation operators. The parameters for these operators are given in Table 5.4.

<table>
<thead>
<tr>
<th>Table 5.3: Common Parameters and Values for Coastal City Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Hazard Events</td>
</tr>
<tr>
<td>Event 1 Scenarios</td>
</tr>
<tr>
<td>Event 2 Scenarios</td>
</tr>
<tr>
<td>Components</td>
</tr>
<tr>
<td>Road</td>
</tr>
<tr>
<td>Levee</td>
</tr>
<tr>
<td>Improvement Choices</td>
</tr>
<tr>
<td>Objectives</td>
</tr>
<tr>
<td>Decision Variables</td>
</tr>
<tr>
<td>Number of Runs</td>
</tr>
<tr>
<td>Generations</td>
</tr>
<tr>
<td>Population Size</td>
</tr>
</tbody>
</table>
For NSGA–II, the elitism operator selected 100 elite solutions, equal to the size of the population. EMODS was run with an archive size of 100, with 50 of the archive slots for nondominated solutions and 50 of the slots reserved for diversity solutions. EMODS was also run at four different relaxation levels for each experiment group: 10%, 5%, 2.5% and 0%. The variants are reported as EMODS$_{10\%}$, EMODS$_{5\%}$, EMODS$_{2.5\%}$, and EMODS$_{ND}$, respectively. The final setting was run without diversity operators. Thus, only the first set of criteria was used for the selection operator, and $t_D$ was set to 0 while $t_N$ was set to 100.

EMODS, NSGA–II and the random search were implemented in C and C++, and the CIS simulation and resilience metric calculation framework was implemented in C++. An evaluation of a CIS improvement plan is somewhat intensive for the chosen number of scenarios, taking around one minute on average for Experimental Group 2, with four objectives. As a result, a distributed computing framework was used to reduce the time for each run. The different methods and resilience metric calculation framework were linked using Optimus Prime, a parallel metaheuristic optimization framework for linking population–based optimization methods with scientific simulations (Sreepathi and Mahinthakumar, 2012). The runs were performed on a cluster with cores that run at 1.4GHz and have 64GB of RAM on each node. Each method was run on 100 cores, which resulted in one CIS improvement plan evaluation per core per generation.

### 5.5.2 Experimental Group 1 Results: Two–Objective Problem

The results of the performance metrics (hypervolume, $ADNO$, $AM$, $ADNN$) for Experimental Group 1 are shown in Figure 5.8. Overall, the results show that greater relaxation decreased hypervolume re-
results, but increased the diversity metric results. Additionally, including diversity operators (archiving and selection) has a large effect on the level of diversity EMODS found. Both EMODS\textsubscript{ND} and NSGA–II have diversity results that are greatly below those obtained by EMODS with relaxations above 0%.

The random search is excluded from the plots in Figure 5.8. In later tables where the actual numbers are presented, it will be shown that random search consistently resulted in extremely high diversity, but such low hypervolume that diversity and hypervolume results are not comparable.

The average and standard deviation of the hypervolume are shown in Table 5.5. As previously noted, as the relaxation parameter decreases, the hypervolume result increases for EMODS. This indicates that the relaxation parameter can be adjusted to control the selection pressure for nondominated solutions. In this Experimental Group (Experimental Group 1) NSGA–II achieves a slightly
Table 5.5: Hypervolume Metric Results for Experimental Group 1 (15 Trials)

<table>
<thead>
<tr>
<th>Method</th>
<th>Hypervolume</th>
<th>Nondominated Front size</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMODS&lt;sub&gt;2.5%&lt;/sub&gt;</td>
<td>3.353 ± 0.03</td>
<td>47.6 ± 3.574</td>
</tr>
<tr>
<td>EMODS&lt;sub&gt;5%&lt;/sub&gt;</td>
<td>3.350 ± 0.005</td>
<td>44.733 ± 6.115</td>
</tr>
<tr>
<td>EMODS&lt;sub&gt;10%&lt;/sub&gt;</td>
<td>3.343 ± 0.004</td>
<td>32.667 ± 6.518</td>
</tr>
<tr>
<td>EMODS&lt;sub&gt;ND&lt;/sub&gt;</td>
<td>3.385 ± 0.006</td>
<td>50.0 ± 0.0</td>
</tr>
<tr>
<td>NSGA–II</td>
<td>3.392 ± 0.005</td>
<td>50.0 ± 0.0</td>
</tr>
<tr>
<td>Random Search</td>
<td>3.071 ± 0.008</td>
<td>21.8 ± 4.75</td>
</tr>
</tbody>
</table>

higher average hypervolume than EMODS<sub>ND</sub> obtains. The table also shows how poorly random search performed compared to the other methods.

The mix of pressures on nondominated solutions and diversity is shown in the final column of Table 5.5. The column shows that when relaxation is allowed, EMODS may not achieve a nondominated front of size 50. Note also that EMODS only had a target size of 100 for EMODS<sub>10%</sub>, EMODS<sub>5%</sub>, and EMODS<sub>2.5%</sub>, whereas the target size for elites was 100 for EMODS<sub>ND</sub> and NSGA–II.

To get a better idea of what the differences in hypervolume reflect for this two–objective problem, final solution sets from a single run of each method (except random search) are displayed in Figure 5.9. In these figures, the serviceability metric was converted to a minimization by subtracting it from 1, so the ideal point is at the lower right of the plots. The final front from EMODS<sub>10%</sub>, in Figure 5.9a, shows that a 10% relaxation EMODS has less well–defined Pareto front than other final fronts. As relaxation decreases to 5% and 2.5%, as in Figures 5.9b and 5.9c, a better–defined Pareto front forms in the lower–right of the plotted areas. The reason for better hypervolume numbers from NSGA–II than from EMODS<sub>ND</sub> is shown by comparing Figure 5.9d to Figure 5.9e. In these plots, it shows that the final front located by NSGA–II contains more extreme solutions than those found by EMODS<sub>ND</sub>.

Additionally, Figure 5.9a shows that with 10% relaxation, diversity solutions represent a broad range of objective function values. In Figures 5.9b and 5.9c with 5% and 2.5% relaxations respectively,
Figure 5.9: Final Archives and Populations from each EMO for Experimental Group 1. Depicts objective function values for nondominated solutions (red solutions), nearly-nondominated diversity solutions (blue), and final populations not in the archive (green solutions). Serviceability, on the vertical axis, has been converted to a minimization by taking 1 - Serviceability for each solution. The ideal point for each graph is in the lower left.
solutions had less diversity, and the diverse solutions that were found were tended to have higher cost and greater serviceability (at the bottom–right of the nondominated fronts).

The results of the diversity metrics are shown in Table 5.6. The results show that there is twice as much average decision distance with just 2.5% relaxation compared to the EMODS_{ND} and NSGA–II runs, for each diversity metric. Comparing the $ADNN$ and $ADNO$ metrics shows that the diversity versions of EMODS obtain nearly–nondominated solutions that are more diverse from nondominated solutions than from other nearly–nondominated solutions. The comparison is slightly complicated by the observation that as relaxation decreases, nearly–nondominated solutions seem to become more concentrated in certain areas of the objective space (as indicated in the graphs in Figure 5.9). Nonetheless, with this potential issue, each of the relaxed versions of EMODS achieve higher $ADNN$ than $ADNO$ metrics, showing that nearly–nondominated solutions can provide highly diverse solutions for the obtained nondominated solutions.

Additionally, note that the $ADNO$ and $ADNN$ metrics are identical for EMODS_{ND}, which is expected since the archiving procedure for this method will only accept nondominated solutions.

Overall, the results of EMODS on the two–objective problems in Experimental Group 1 are of high–quality in the objective and especially the decision spaces. While EMODS does seem to concentrate on specific regions of the objective space for obtaining nearly–nondominated solutions, earlier results in Piper and Ranjithan (2013) suggest that this behavior is due to characteristics of the

<table>
<thead>
<tr>
<th>Method</th>
<th>$ADNO$</th>
<th>$AM$</th>
<th>$ADNN$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMODS_{10%}</td>
<td>812.796 ± 75.960</td>
<td>464.377 ± 77.878</td>
<td>838.691 ± 39.935</td>
</tr>
<tr>
<td>EMODS_{5%}</td>
<td>769.146 ± 56.466</td>
<td>352.886 ± 56.575</td>
<td>782.450 ± 56.575</td>
</tr>
<tr>
<td>EMODS_{2.5%}</td>
<td>727.664 ± 42.397</td>
<td>342.006 ± 41.610</td>
<td>739.471 ± 46.673</td>
</tr>
<tr>
<td>EMODS_{ND}</td>
<td>328.291 ± 32.660</td>
<td>114.522 ± 7.762</td>
<td>328.291 ± 32.660</td>
</tr>
<tr>
<td>NSGA–II</td>
<td>353.707 ± 38.149</td>
<td>114.818 ± 9.146</td>
<td>353.7567 ± 36.863</td>
</tr>
<tr>
<td>Random Search</td>
<td>1823.204 ± 74.311</td>
<td>1756.80 ± 68.195</td>
<td>1823.204 ± 74.311</td>
</tr>
</tbody>
</table>
5.5.3 Experimental Group 2 Results: Four-Objective Problem

The results from the runs for the four objective problem are given in this section. The averages in this section are calculated for only five randomly seeded runs rather than fifteen as in Experimental Group 1. The reduced number of runs was chosen because each run in this group took approximately eight hours, due to the increased computational complexity of the serviceability recovery time and demand satisfaction resilience metrics.

Graphs comparing the methods on all performance metrics are shown in Figure 5.10. As with Experimental Group 1, Figure 5.10a shows that hypervolume decreased as relaxation parameter increased. NSGA-II is known to perform worse as the number objectives increases beyond three, and EMODS\textsubscript{ND} seems to generally provide a better nondominated set according to the hypervolume metric.

Somewhat surprisingly, most of the diversity metrics show relatively similar performance across different relaxation levels for EMODS. The \textit{ADNO} metric comparison in Figure 5.10b shows a slight decrease in the \textit{ADNO} metric as the relaxation increases from 2.5% to 10%. That this result seems counter-intuitive could be partly explained by the lower number of runs that was performed. Additionally, the relative absence of conflict between certain metrics noted in Piper et al. (2013a) may result in necessary regions of the decision and objective space being preserved.

Perhaps most surprisingly, the \textit{ADNN} metric, compared in Figure 5.10d, shows that the \textit{ADNN} metric for EMODS with diversity operators was lowest with a 5% relaxation, and higher with both 2.5% and 10%. The presence of outliers both high and low for the runs with 5% relaxation indicate that five runs may be insufficient to draw strong conclusions.

The average and standard deviation of hypervolume obtained by each method are shown in Table 5.7. In general, EMODS with diversity operators fared worse than NSGA-II in terms of convergence and spread for its final nondominated set approximation. Without diversity operators,
Figure 5.10: Comparison of Performance Metrics (Hypervolume, \textit{ADNO}, \textit{AM}, and \textit{ADNN}) by Method from 15 Trials for Experimental Group 2. For each metric, higher is better.
EMODS\textsubscript{$ND$} had slightly higher hypervolume than NSGA–II. The variants of EMODS that included diversity operators achieved reasonable hypervolume performance, however, indicating again that the relaxation parameter is likely to affect convergence.

As before, it should be noted that the hypervolume numbers achieved by EMODS\textsubscript{10\%}, EMODS\textsubscript{5\%}, and EMODS\textsubscript{2.5\%} are obtained while only seeking half as many nondominated solutions as EMODS\textsubscript{$ND$} and NSGA–II. With four objectives, however, there are more options for solutions to be nondominated. As a result, each EMODS method had a final nondominated front of at least size 50 in this Experimental Group. It should also be noted that all EMO methods were significantly better than random search in hypervolume performance.

The average hypervolume for each generation and method is shown in Figure 5.11. The graph demonstrates that until around generation 150, EMODS\textsubscript{$ND$} and NSGA–II are similar, but EMODS\textsubscript{$ND$} is generally higher than NSGA–II after about 150 iterations. The gaps among EMODS with diversity operators stays relatively steady. The performance advantage of the non–diversity seeking methods stays consistent over EMODS variants that include diversity operators.

A concern other studies have raised about hypervolume–based evolutionary algorithms is that the complexity of hypervolume diminishes the applicability of hypervolume for problems with more than two or three objectives. While these experiments are not meant to exclusively address hypervolume computation time, some observations can be made. The average computation times for each

<table>
<thead>
<tr>
<th>Method</th>
<th>Hypervolume</th>
<th>Nondominated Front Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMODS\textsubscript{10%}</td>
<td>9.190 ± 0.047</td>
<td>50.0 ± 0.0</td>
</tr>
<tr>
<td>EMODS\textsubscript{5%}</td>
<td>9.217 ± 0.021</td>
<td>50.0 ± 0.0</td>
</tr>
<tr>
<td>EMODS\textsubscript{2.5%}</td>
<td>9.271 ± 0.029</td>
<td>50.0 ± 0.0</td>
</tr>
<tr>
<td>EMODS\textsubscript{$ND$}</td>
<td>9.416 ± 0.034</td>
<td>50.0 ± 0.0</td>
</tr>
<tr>
<td>NSGA–II</td>
<td>9.367 ± 0.026</td>
<td>50.0 ± 0.0</td>
</tr>
<tr>
<td>Random Search</td>
<td>7.855 ± 0.241</td>
<td>49.6 ± 0.8</td>
</tr>
</tbody>
</table>
Figure 5.11: Average Hypervolume Convergence of each Method for Experimental Group 2. Each line represents the average hypervolume over five runs at that iteration for each method (excluding random search). Wider bands around each method indicate higher variability in the hypervolume at that iteration.
iteration and each method are compared in Figure 5.12. In this graph, computation time per iteration includes the total time required for the machine to process the iteration, but does not include fitness evaluation time.

The graph shows that EMODS\(_{ND}\) takes the longest time each iteration. The cause may be that EMODS\(_{ND}\) needs longer to calculate hypervolume contributions to trim its archive to under 100 solutions. The result is that more exclusive hypervolume contributions have to be performed than the variants of EMODS with diversity operators.

The increased computational time spent could be seen as an argument against using EMODS for a problem with this many objectives. A counterpoint is presented when the algorithms’ times for each iteration are compared against the fitness times for each iteration, which is shown in Fig-
Figure 5.13: Average Computation Time per Iteration of Each Method and Fitness Evaluation Time. Wider bands around each method indicate higher variability in the computation time at that iteration.

Figure 5.13. The graph shows that while EMODS, particularly EMODS\textsubscript{ND}, is slower each iteration than NSGA–II, fitness function evaluations dominate the computational time in each iteration. For the current problem, speeding up the CIS improvement plan evaluations is likely to be far more beneficial than choosing a faster EMO.

To illustrate this, consider the following. At generation 500, NSGA–II may be finishing each iteration almost two seconds faster, but due to the fitness function evaluation time, it takes thirty iterations of EMODS\textsubscript{ND} for NSGA–II to gain an iteration, or approximately thirty minutes. After an hour, NSGA–II would gain two iterations, and over twenty–four hours of running, NSGA–II would gain a bit more than twenty–four iterations, during which time EMODS\textsubscript{ND} could have done approximately...
Table 5.8: Diversity Metric Results for Experimental Group 2 (5 Trials)

<table>
<thead>
<tr>
<th>Method</th>
<th>$ADNO$</th>
<th>$AM$</th>
<th>$ADNN$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMODS$_{10%}$</td>
<td>862.894 ± 69.990</td>
<td>543.622 ± 85.814</td>
<td>946.728 ± 42.322</td>
</tr>
<tr>
<td>EMODS$_{5%}$</td>
<td>852.710 ± 48.876</td>
<td>540.090 ± 35.744</td>
<td>859.844 ± 37.282</td>
</tr>
<tr>
<td>EMODS$_{2.5%}$</td>
<td>887.754 ± 40.582</td>
<td>538.494 ± 30.886</td>
<td>930.062 ± 32.467</td>
</tr>
<tr>
<td>EMODS$_{ND}$</td>
<td>458.102 ± 31.038</td>
<td>261.750 ± 15.560</td>
<td>458.102 ± 31.038</td>
</tr>
<tr>
<td>NSGA–II</td>
<td>473.504 ± 36.553</td>
<td>214.694 ± 11.833</td>
<td>473.504 ± 36.553</td>
</tr>
<tr>
<td>Random Search</td>
<td>2047.266 ± 5.729</td>
<td>1934.809 ± 8.228</td>
<td>2047.266 ± 5.729</td>
</tr>
</tbody>
</table>

1300 iterations at sixty–six seconds each. The twenty–four iterations, while potentially useful, are still small compared to the total number of iterations that both methods would have.

The results of diversity metrics for Experimental Group 2 are summarized in Table 5.8. As with Experimental Group 1, the EMODS variants using diversity operators almost always have twice the diversity metric values than EMODS$_{ND}$ or NSGA–II. Again, $ADNN$ is slightly higher than $ADNO$, showing that EMODS finds distinct, diverse, and substitutable alternatives to the nondominated solutions that are obtained. For both EMODS$_{ND}$ and NSGA–II, the $ADNO$ and $ADNN$ metrics are identical, which shows that all solutions in all final archives produced by each method were nondominated.

Correlations were calculated for the final solution values for each run of the EMO methods. The correlations are shown in Table 5.9, which shows a strong positive correlation between demand satisfaction and serviceability metrics. Overall, few of the metrics consistently conflict, except each resilience metric conflicts with cost. Due to the strong agreement between demand satisfaction and serviceability, demand satisfaction will be dropped when the improvement plans are evaluated for the secondary set of hazard scenarios.

### 5.5.4 Evaluation of Solution Robustness

To evaluate the robustness of solutions, an alternative set of hazard scenarios was generated and used to test. Again, sixty–two hazard scenarios were created and divided into two stages, with one
Table 5.9: Correlations between Metrics for Experimental Group 2

<table>
<thead>
<tr>
<th>Objectives</th>
<th>Serviceability</th>
<th>Restoration Time</th>
<th>Demand Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>-0.3125</td>
<td>-0.5448</td>
<td>-0.2182</td>
</tr>
<tr>
<td>Serviceability</td>
<td>-</td>
<td>0.5217</td>
<td>0.9783</td>
</tr>
<tr>
<td>Recovery Time</td>
<td>-</td>
<td>-</td>
<td>0.4338</td>
</tr>
</tbody>
</table>

scenario in each stage being the no–hazard scenario. Each of the improvement plans obtained in Experimental Group 1 was evaluated on this new hazard scenario set, which will be called “Alternative Hazard Scenarios” (AHS). The improvement plan costs are unchanged, only the resilience metrics change values. For AHS, only serviceability and serviceability restoration time will be calculated. The correlations between metrics in Table 5.9 provides evidence that the high correlation between demand satisfaction and serviceability for the Coastal City CIS permits considering only one of these metrics. The demand satisfaction metric is not chosen because the serviceability metric is more computationally tractable. The analysis of these resilience metrics on AHS is reflective of the real–world situation, where forecasting errors will result in the improvement plans being optimized for hazard scenarios that are unlikely to occur precisely as predicted.

Each improvement plan obtained by the final archives from Experimental Group 1 by the EMO methods was evaluated for serviceability and serviceability restoration time in the AHS. Cost was then added and a nondominated sorting determined the new nondominated front for the resulting objective function values. Overall results for the post–analysis are shown in Table 5.10. The column \textit{Old Front Size} indicates the average size of the fronts obtained in Experimental Group 1 for each method. \textit{New Front Size} indicates the size of the nondominated front with the new evaluations of the resilience metrics. The columns \textit{Change} and \textit{% Change} show the change and percent change in front sizes. The final column, \textit{Nearly–Nondominated Additions}, shows the average number of nearly–nondominated solutions that became nondominated solutions in AHS.

While all methods had a decrease in average front sizes, the table shows that the decrease was smaller for diversity seeking methods. Additionally, under AHS, each final archive of diversity seek-
Table 5.10: Change in Size of Nondominated Fronts under Alternative Hazard Scenarios

<table>
<thead>
<tr>
<th>Method</th>
<th>Old Front Size</th>
<th>New Front Size</th>
<th>Change</th>
<th>% Change</th>
<th>Nearly–Nondominated Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMODS2.5%</td>
<td>47.60</td>
<td>39.27</td>
<td>-8.33</td>
<td>-17.7%</td>
<td>3.53</td>
</tr>
<tr>
<td>EMODS5%</td>
<td>44.73</td>
<td>35.80</td>
<td>-8.93</td>
<td>-18.9%</td>
<td>3.4</td>
</tr>
<tr>
<td>EMODS10%</td>
<td>32.67</td>
<td>28.93</td>
<td>-3.73</td>
<td>-11.4%</td>
<td>4.06</td>
</tr>
<tr>
<td>EMODSND</td>
<td>86.47</td>
<td>59.80</td>
<td>-26.67</td>
<td>-30.7%</td>
<td>0</td>
</tr>
<tr>
<td>NSGA–II</td>
<td>96.00</td>
<td>68.73</td>
<td>-27.27</td>
<td>-28.3%</td>
<td>0</td>
</tr>
</tbody>
</table>

ing methods had an average of three to four formerly nearly–nondominated solutions that became nondominated. This indicates that diversity had at least a small, but direct, effect on the reduced decreases of nondominated front sizes. For further evidence, Figure 5.14 shows that for a few instances, EMODS5% and EMODS10% increased the size of the nondominated front under AHS.

The diversity metric values for the nearly–nondominated additions that become nondominated solutions under AHS can be compared with the diversity metric values for nondominated solutions and former nondominated solutions. The comparison for the ADNO diversity metric is shown in Figure 5.15. The figure contains average ADNO values for EMODS5% and EMODSND over three groups of solutions: Formerly nondominated solutions (no longer nondominated in the AHS), Former nearly–nondominated solutions (nondominated in the AHS), and nondominated solutions that remain nondominated solutions in the AHS.

Figure 5.15 shows that the nearly–nondominated solutions that became nondominated in the Alternative Hazard Scenario had high diversity on average. This demonstrates that distinct and substitutable solutions are connected to the ability of a set of solutions to be robust to different hazard scenarios. The is large variability in the diversity of the former nearly–nondominated solutions suggest that this is not always true.
Figure 5.14: Change in Size of Nondominated Fronts under Alternative Hazard Scenarios. The vertical axis gives the change if the size of the new front (the front formed after the evaluation of solutions in the alternative hazard scenario) subtracts the size of the old front (the front found during Experimental Group 1). Higher values are better.
Figure 5.15: ADNO Diversity Metrics For Solutions from Final Archives. Both the EMODS$_{5\%}$ (red, three bars on the left) and EMODS$_{ND}$ (blue, three bars on the right) are divided into three groups, as noted in the text. The three groups of solutions are Formerly nondominated solutions (no longer nondominated in the AHS), Former nearly-nondominated solutions (nondominated in the AHS), and nondominated solutions that remain nondominated solutions in the AHS. EMODS$_{ND}$ has no nearly-nondominated solutions in the archive, thus this group has 0 ADNO value.
Table 5.11: Objective Function Values of Selected Solutions

<table>
<thead>
<tr>
<th>Solution</th>
<th>Cost</th>
<th>Serviceability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.01E6</td>
<td>74.85%</td>
</tr>
<tr>
<td>2</td>
<td>4.06E6</td>
<td>74.87%</td>
</tr>
<tr>
<td>3</td>
<td>4.40E6</td>
<td>74.59%</td>
</tr>
<tr>
<td>4</td>
<td>4.35E6</td>
<td>74.58%</td>
</tr>
<tr>
<td>5</td>
<td>4.43E6</td>
<td>74.60%</td>
</tr>
</tbody>
</table>

5.5.5 Comparison of Selected Solutions

To illustrate the diversity of solutions that EMODS is able to locate, a few solutions from Experimental Group 1 (two–objective problem) were examined in greater detail. Solution 1 (S1) is a nondominated solution from a run of EMODS_{ND}. Solution 2 (S2) is from the same run of EMODS_{ND}, and it is the closest nondominated solution in the objective space to S1. Solutions 3, 4, and 5 (S3, S4, and S5) are from the final archive of a single run of EMODS_{ND}. S3 and S4 are both nondominated solutions, and S4 is one of the closest solutions to S3 in the objective space. S5 is one of the closest nearly–nondominated, diversity solutions to S3. Additionally, S5 becomes a nondominated solution in AHS. The (Experimental Group 1) objective function values of all of these solutions are given in Table 5.11.

The first stage improvements on the Coastal City infrastructures for S1 and S2 are displayed in Figure 5.16, and the same class of improvements for S3, S4, S5 is displayed in Figure 5.17. S1 (Figure 5.16a) and S2 (Figure 5.16b) appear to have many of the same decisions, which is more apparent in the center of the maps, away from the coastal areas. The solutions S3 (Figure 5.17a) and S4 (Figure 5.17b) appear to have a greater number of differences. For instance, sections of the road network near the center of the the Coastal City region in these two solutions show many differences. S5 (Figure 5.17c) shows a similar number of differences to S3 and S4. The levee improvements chosen, however, are similar for the selected solutions. This suggests that improving the levees is critical for raising the overall resilience of Coastal City CIS. Additionally, a road segment connecting the downtown (lower–right) region of Coastal City to a main roadway running WNW to ESE receives an level
two improvement in every selected solution, an observation that would be important for a decision-maker.

The decision space distances among these solutions reflect many of the qualitative observations of solution differences noted above. The decision space distance between $S1$ and $S2$ is 291 according to the distance metric defined in Equation (5.7). Compared to the mean in Table 5.6, this decision distance is slightly lower than the average $ADNO$ and $ADNN$ metrics achieved with this method.

The decision distance from $S3$ to solutions $S4$ and $S5$, however, is 648 and 588, respectively. Again, this is lower than the average $ADNO$ and $ADNN$ metrics reported in Table 5.6, yet both distances are higher than the decision distance between solutions $S1$ and $S2$. In this instance, the decision distance between nondominated solutions $S3$ and $S4$ is even greater than the decision distance between the nondominated solution $S3$ and the nearly-nondominated and diverse solution $S5$. The pursuit of diversity in the decision space seems to increase diversity even in the nondominated solutions obtained by the final archive.

5.6 Observations

Improving resilience for the CIS against natural hazards is a multi-faceted challenge. In this paper, a multiobjective approach was taken to address various aspects of CIS resilience. A modeling framework was introduced to quantify aspects of resilience and capture the effects of interdependencies among CIS components when natural hazards reduce lifeline services that components provide. The modeling framework is necessarily incomplete due to uncertainties in the future hazards and inability to quantify certain aspects of the CIS.

The EMODS evolutionary algorithm was applied to the optimization of CIS improvement plans in Coastal City to provide a multiobjective optimization method that could simultaneously approximate the Pareto front and find nearly-nondominated solutions distinct in the decision space. EMODS was originally applied to continuous variable optimization problems. As a result, the algorithm was modified to allow it to handle any problem representation where a decision space distance between
Figure 5.16: Stage 1 Improvement plans from EMODS

Legend
- Road Network Improvements
- Stage 1, Level 1 Improvements
- Stage 1, Level 2 Improvements
- Levee System Improvements
- Green components are type 1 upgrades
- Red components are type 2 upgrades
- Components with no color have “no-upgrade” selected.
Figure 5.17: Stage 1 improvement plans from EMODS_{5\%}. Green components are type 1 upgrades and red components are type 2 upgrades. Components with no color have “no-upgrade” selected.
solutions could be defined. This modification expands the problems to which EMODS can be applied and improves the flexibility of the algorithm. The archiving method used in EMODS can be used in other evolutionary algorithms as a fast and effective way to select maximally different subsets of solutions.

The numerical experiments show the modified EMODS algorithm is capable of finding highly diverse solutions to the CIS resilience optimization problem. Retaining relaxed, nearly-nondominated solutions during the optimization process does seem to negatively affect the convergence of EMODS to a nondominated front. Despite this, reasonable nondominated solutions can be found using the algorithm and a properly chosen relaxation parameter. The results from the EMODS_{ND} method show that the basic selection and archiving procedure are at least as effective as NSGA–II in locating a nondominated front for two-objective problems. On problems with a greater number of objective functions, EMODS is potentially more effective.

The robustness of potential improvement plans was evaluated by calculating resilience metric values for the improvement plans obtained in Experimental Group 1 on the AHS. In real world problems, the actual scenarios that occur will be unknown, so any optimization model must provide robust improvement plans. Additionally, the high diversity of the nearly-nondominated solutions that become nondominated in the AHS show that decision diversity is potentially useful in achieving robustness. Thus, diversity pursuing optimization algorithms like EMODS can offer an alternative method to achieve robustness. The examination of selected solutions found in the final archive of EMODS_{ND} and EMODS_{5%} shows that using diversity operators increases the diversity of solutions even on the nondominated front. Further work can be done to improve the decision space exploration of evolutionary algorithms, which can then improve the decision diversity that an algorithm like EMODS achieves.
Observations

The research presented in this dissertation is intended to provide decision-makers a quantitative framework for improving performance, especially resilience, of complex and interdependent systems. The modeling framework that is provided in Chapter 2 is adaptable to a variety of civil infrastructure system (CIS) configurations. Moreover, the ability of the framework to explicitly consider interdependencies when making pre-disaster improvements is a novel, but critical, feature lacking in nearly all other CIS optimization models. The quantification of resilience through the consideration of lifeline services that CIS components provide requires the development of various resilience metrics, some of which are illustrated in this dissertation. These are not the only resilience metrics, and the potential lack of conflict among the metrics that were defined indicates that pursuing and investigating other metrics could be valuable for future research.

The research demonstrates that explicit modeling of interdependencies is possible. One issue is that the ability to model interdependencies might be limited if the optimization models are confined to a mathematical programming approach. Additionally, the research demonstrates that system performance with a large number of possible future scenarios may increase computational complexity,
but the value gained from inclusion of a scenario–based approach suggests that it is worthwhile. As better physical models for predicting CIS damage from storm surge and other natural hazards become available, the ability to use models such as that developed in the modeling framework presented in Chapter 2 and expanded in Chapter 3 will be increased.

While considering multi–stage decisions simultaneously increases the complexity of optimization, Chapter 3 shows that this approach can be worthwhile and potentially computationally tractable when modeled as a stochastic programming problem. The value of the stochastic solution for the metrics shows that it would be preferable to consider as many potential hazard scenarios as possible. A stochastic programming problem has scaling issues, however, especially with regards to the number of hazard scenarios and the number of potential improvement decisions. While special algorithmic approaches are potentially available, each approach potentially requires a new implementation to optimize for each resilience metric. These issues motivate the development of a meta–heuristic approach using an evolutionary multiobjective optimization algorithm, EMODS, which is potentially better able to handle issues of scale and quickly switch between the choice of considered metrics.

Based on the ideas of modeling to generate alternatives (Brill, 1979), EMODS has an additional goal of generating highly diverse solutions to find robust improvement plans. Uncertainties in forecasting storms and impact on the CIS, the state of the CIS, and effectiveness of available technologies result in any model of CIS resilience improvement being highly incomplete. Some aspects will be unrepresented or, like political will, unquantifiable. Providing decision makers with a highly diverse set of alternatives can help consider the effects of some of these uncertainties. Unfortunately, the utility of locating alternatives may be somewhat problem–dependent. For certain problems, alternative solutions may not be sufficiently different to warrant a decision–maker’s attention. It is also possible that uncertainties are sufficiently large so that alternatives would effectively have to optimize an entirely different problem to be useful.

EMODS is shown to be effective on test problems with two and three objectives in Chapter 4, both in terms of achieving a high–quality nondominated front and locating distinct and substi–
tutable solutions. For these test problems, the computational burden from the complexity of the hypervolume–based fitness method seems minimal. The test problems, however, are somewhat “regular” and potentially form a poor representation of the characteristics of real–world multiobjective problems. The different diversity metrics presented in Chapter 4 illustrate the two intended aspects of diversity: spread and substitutability. The need to pursue both of these aspects to achieve diversity is also illustrated in that chapter.

When EMODS was applied to multiobjective resilience problems of the Coastal City CIS in Chapter 5, certain aspects of the resilience metrics and the CIS resilience optimization problem were revealed. For instance, Chapter 2 showed indications that the resilience metrics might not strongly conflict with one another. It could be argued from this point that the resilience metrics presented do not represent necessarily different aspects of resilience. When EMODS was used to optimize the four objective function problem in Chapter 5, there was a great deal of agreement between the demand satisfaction and serviceability resilience metrics in the final archive sets of solutions obtained by evolutionary algorithms. Further investigation is necessary to reveal if the correlation is a product of the Coastal City data, or from the actual structure of these two resilience metrics. Regardless of the source, it makes it possible, if desired, to optimize one aspect of the resilience of Coastal City’s CIS according to either of these metrics.

Chapter 5 demonstrated that in general multiobjective evolutionary algorithms make a viable method for determining improvement strategies for systems subject to multi–event hazards according to various criteria. Additionally, the results from that chapter show that a multiobjective evolutionary algorithm can be inspired by the modeling to generate alternatives approach and can produce reasonable solutions to a real world problem. The ease with which different resilience metrics can be exchanged and adapted further enforces the general applicability of this approach.

Overall, EMODS performed competitively with the NSGA–II evolutionary algorithm on the Coastal City CIS resilience optimization problem. Convergence to a nondominated front exhibits sensitivity to the chosen relaxation parameter, requiring some care in parameter selection. A modification of the EMODS fitness evaluation method and selection criteria could improve this convergence with-
out losing the emphasis on diversity. The most idealistic goal might be a version of EMODS that could automatically adapt the relaxation parameter and diversity pressure based on convergence difficulty and the decision diversity a particular problem exhibits. For example, if a problem has little diversity, the automatically adapting version of EMODS would focus on nondominated front convergence.

EMODS with diversity operators is able to add some robustness to the final solution archives. Chapter 5 showed that errors in forecasting, in the form of alternative hazard scenarios, tends to reduce the number of nondominated solutions that the optimization algorithm found. When EMODS uses diversity operators, however, the drop in the number of nondominated solutions can be lessened and potentially reversed. While diversity seems to be partly responsible for this, a future experiment could further investigate this link.

The proposed algorithm and modeling framework are potential tools for improving the resilience of evolving, complex systems such as the CIS. Future efforts must expand the ability of CIS modeling frameworks to better incorporate the results of simulated and forecasted natural hazards. Additionally, domain expertise needs to be leveraged to create mechanisms that capture the interdependencies among the various subsystems of the CIS. The features of the modeling framework presented in this dissertation enable decision–makers to consider the system–wide effects of improvements to a CIS system and improve the resilience of the CIS against a range of natural hazards. Application of the framework, however, requires expertise in modeling and computing. One of the most promising and critical avenues for future research in CIS resilience is to develop decision analysis tools that are easy to implement and use to provide decision–makers with the tools needed to devise better ways to protect their systems. The research in this dissertation can serve as a launching point for building these decision analysis tools.
REFERENCES


161


