SANDIA REPORT
SAND2013-7755
Unlimited Release
Printed September 2013

A Modeling Framework for Investment Planning in Interdependent Infrastructures in Multi-Hazard Environments

Nathanael J. K. Brown, Jared L. Gearhart, Dean A. Jones, Linda K. Nozick, and Michael Prince

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-7755
Unlimited Release
Printed September 2013

A Modeling Framework for Investment Planning in Interdependent Infrastructures in Multi-Hazard Environments

Nathanael J. K. Brown, Jared L. Gearhart, Dean A. Jones, Linda K. Nozick, and Michael Prince
Department 06131, Operations Research and Computational Analysis (ORCA)
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1188

Abstract

Currently, much of protection planning is conducted separately for each infrastructure and hazard. Limited funding requires a balance of expenditures between terrorism and natural hazards based on potential impacts. This report documents the results of a Laboratory Directed Research & Development (LDRD) project that created a modeling framework for investment planning in interdependent infrastructures focused on multiple hazards, including terrorism. To develop this framework, three modeling elements were integrated: natural hazards, terrorism, and interdependent infrastructures. For natural hazards, a methodology was created for specifying events consistent with regional hazards. For terrorism, we modeled the terrorist's actions based on assumptions regarding their knowledge, goals, and target identification strategy. For infrastructures, we focused on predicting post-event performance due to specific terrorist attacks and natural hazard events, tempered by appropriate infrastructure investments. We demonstrate the utility of this framework with various examples, including protection of electric power, roadway, and hospital networks.

ACKNOWLEDGMENTS

The authors would like to thank the following Sandia National Laboratories staff for making various contributions to the project: Nathaniel Martin, Katherine Jones, Alisa Bandlow, Rich Detry, and Mercy DeMenno. Additionally, we would like to acknowledge the outstanding technical partnership with the following students from Cornell University who helped make this work possible: Anna Li, Natalia Romero, and Ningxiong Xu.

This work was supported by Laboratory Directed Research and Development funding from Sandia National Laboratories.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

CSV	Comma-Separated Values
DHS	Department of Homeland Security
DOE	Department of Energy
DTA	Dynamic Traffic Analysis
FEMA	Federal Emergency Management Agency
GA	Genetic Algorithm
HPC	High Performance Computing
LDRD	Laboratory Directed Research & Development
LP	Linear Program
MILP	Mixed Integer Linear Program
MPI	Message Passing Interface
NMSZ	New Madrid Seismic Zone
OD	Origin-Destination (table of network flow values)
OPL	Optimization Programming Language
RMI	Remote Method Invocation
SNL	Sandia National Laboratories
TAZ	Traffic Analysis Zone
XML	eXtensible Markup Language

1. INTRODUCTION

The continuous operation of infrastructure systems is critical to societal welfare. Much of the current planning for protection against natural hazards and terrorism is done separately for each infrastructure and each hazard, which precludes a cost-benefit analysis across multiple investments. The goal of this project is to create a modeling framework for investment planning in interdependent infrastructures, focusing on multi-hazard risks, including terrorism. With such a framework, it is possible to balance expenditures for terrorism and natural hazards based on potential impacts. To develop the framework, three modeling elements were integrated: natural hazards, terrorism, and interdependent infrastructures. For natural hazards, a methodology was created to specify events that are consistent with regional hazards. For terrorism, sophisticated models of potential terrorist actions were developed based on assumptions regarding their knowledge, goals, and target identification strategy. For infrastructures, our efforts focused on predicting the post-event performance of a system impacted by specific terrorist attacks and natural hazard events, as well as the effect of appropriate infrastructure investments.

The missions of the Department of Energy (DOE) and the Department of Homeland Security (DHS) are to advance the security of the United States through technical means, including energy and infrastructure security, respectively. The tools developed under this project can be applied to support these missions, because they are applicable across a wide range of infrastructures. Further, two key responsibilities of the Department of Homeland Security are to analyze and to mitigate the consequences of national disasters and terrorist events. To define a reasonable scope for this effort, this research was limited to three different infrastructures: the Eastern Interconnect Power Grid, the roadway network in and around the Memphis metropolitan area, and the network of hospitals in the Memphis metropolitan area. Similarly, although multiple natural hazards can be modeled using the techniques developed during this analysis, the focus was solely on the seismic threat present in the New Madrid Seismic Zone (NMSZ).

The first portion of this document describes the capabilities developed during this effort, including the associated applications. Note that most of the capabilities have short summaries that reference full papers, except for the air transport model which was not published externally. The second portion of this document discusses how these capabilities can be combined to produce a comprehensive cost-benefit analysis for investing in two separate infrastructures (roadways and hospitals) to mitigate against two different threats (earthquakes and terrorism).

2. SUMMARY OF CAPABILITIES

During the course of this 3-year LDRD, our team developed a variety of capabilities, which were primarily documented in eight peer-reviewed publications. This section provides a brief summary of these capabilities as well as references to the associated publications which provide more detailed information.

2.1. Modeling of Consequence Scenarios for Natural Disasters

One effective methodology to accurately assess the impact of natural disasters on an infrastructure is to create a collection of consequence scenarios that are consistent with the regional hazard. By leveraging Hazus, the loss estimation methodology developed by the Federal Emergency Management Agency (FEMA), it is possible to use optimization to create a small set of consequence scenarios that accurately represent the hazard more efficiently than using Monte Carlo sampling. For our study, we focused on the effects of earthquakes on roadway bridges and the electric power grid in the NMSZ. However, this technique could also be applied to the effects of hurricanes and could include other infrastructures. A full description of this optimization-based scenario generation is described in Brown, et al. (2011), and extended in Gearhart, et al. (2013).

2.2. Terrorism Model

The modeling of terrorism as a hazard to an infrastructure followed a game-theoretic approach using leader-follower assumptions. Two different approaches were researched during the course of this LDRD project. A Benders Decomposition approach was applied to power network interdiction in the paper by Xu, et al. (2103), "A Decomposition Heuristic for a Power Network Interdiction Problem." A simplified game-theoretic approach is examined in the section of the case study describing the terrorism threat solution procedure.

2.3. Electric Power Grid Model

For the electric power system, post-event impacts are modeled using a DC flow economic dispatch model as well as a cascading power failure model. The models utilized in this project are described both in Romero, et al. (2012), and Romero, et al. (2013).

2.4. Roadway Network Model

Throughout the course of our research, we produced two roadway network models: New Orleans, Louisiana, and Memphis, Tennessee. The collection of files associated with each network describes all road capacities, speeds, bridge locations, and population density. Appendix A describes the input files required to specify a highway network model, as well as the simulation procedure that provides all traffic metrics.

The network performance for these network models is measured using a dynamic traffic assignment (DTA) algorithm. The DTA is the core of the roadway network model, and simulates the movement of a fixed number of vehicles from a collection of origins to a

collection of destinations over a well-defined time period. It was originally developed in MATLAB at Cornell University for modeling the Katrina evacuation. The functionality was ported to a Java implementation at Sandia, and modified so that it can analyze either a roadway network with no bridges or a roadway network with multiple bridges in various damage states (which restricts traffic flow). The model uses Dijkstra's algorithm for determining the shortest path, based on travel time, between all OD pairs. At each time interval, traffic can be re-routed based on changing congestion patters using an "all-or-nothing" selection strategy (i.e., all traffic bound for a particular destination will choose the shortest path to get there). The Sandia version of the code is able to run the Katrina evacuation simulation in roughly 10 seconds on an 8-core PC, as compared to 110 minutes for DynusT (the industry standard) and approximately 24 hours for TRANSIMS. A complete description of the DTA in the context of the Katrina study can be found in Li, et al. (2012).

2.5. Air System Model

The purpose of the air system model is to estimate the resulting effect on passenger travel, given a hazard scenario and a resulting set of airports that are inoperable (either directly from the hazard or potentially due to disruptions to the power grid). More specifically, given a set of inoperable airports, the model estimates the groups of passengers that are no longer able to reach their destination within the given time window. Based on these results, analysis can identify where the air system may be most vulnerable, and investment planning decisions can be made accordingly. Appendix B gives an in-depth description of the air system model including instructions on how to operate the software and input file formats.

2.6. Optimization

In order to utilize the objective values produced by these different models for investment planning, a multi-objective optimization capability was developed. Since many investment planning scenarios have an exceedingly large solution space, computational efficiency is paramount but insufficient. Additional techniques must be applied in order to achieve near-optimal results in a reasonable amount of time. For our research, we made use of heuristic techniques such as Tabu Search and genetic algorithms as well as parallel processing on Sandia's HPC (High Performance Computing) resource Red Sky. For example, the DTA algorithm can be run on a single core, across several threads on a multi-core machine, or across multiple processes on a multi-node supercomputer. When spawned across multiple nodes, MPI (Message Passing Interface) is used to spawn a single process to each node and RMI (Remote Method Invocation) is used to send the results back from each child process to the parent process. The parent process collects the resulting solutions from each child process, produces a new generation of mitigation strategies via genetic crossover, and respawns the new collection of strategies to be evaluated by the child processes. The solutions from each iteration are used to construct a Pareto frontier, which holds the final collection of solutions and avoids arbitrary weighting of the objectives to allow decision makers to assign their own importance. The various pieces of this optimization methodology are described in more detail in the case study.

3. CASE STUDY

This section of the report discusses how our framework could be applied to the core problem of investment planning in interdependent infrastructures in a multi-hazard environment. The two infrastructures selected are the roadway system in Memphis, Tennessee, and the network of hospitals that is connected by those roads. The hazards in this region include the NMSZ seismic threat to the roadway bridges and an act of terrorism against the hospitals. This optimization is an extension of the multi-objective, 2-stage stochastic problem described in Brown, et al. (2013). In that paper, there were two interdependent infrastructures (the Memphis roadway network with bridges and the hospital system), but only a single hazard (seismic). Although it is somewhat unlikely that the Memphis hospital system would be targeted by a terrorist, this case study serves as an illustrative example of how an investment strategy could be evaluated by a decision maker who is concerned with maintaining an operational capability subject to multiple threats. Figure 1, below, illustrates the solution procedure from the decision maker's perspective.

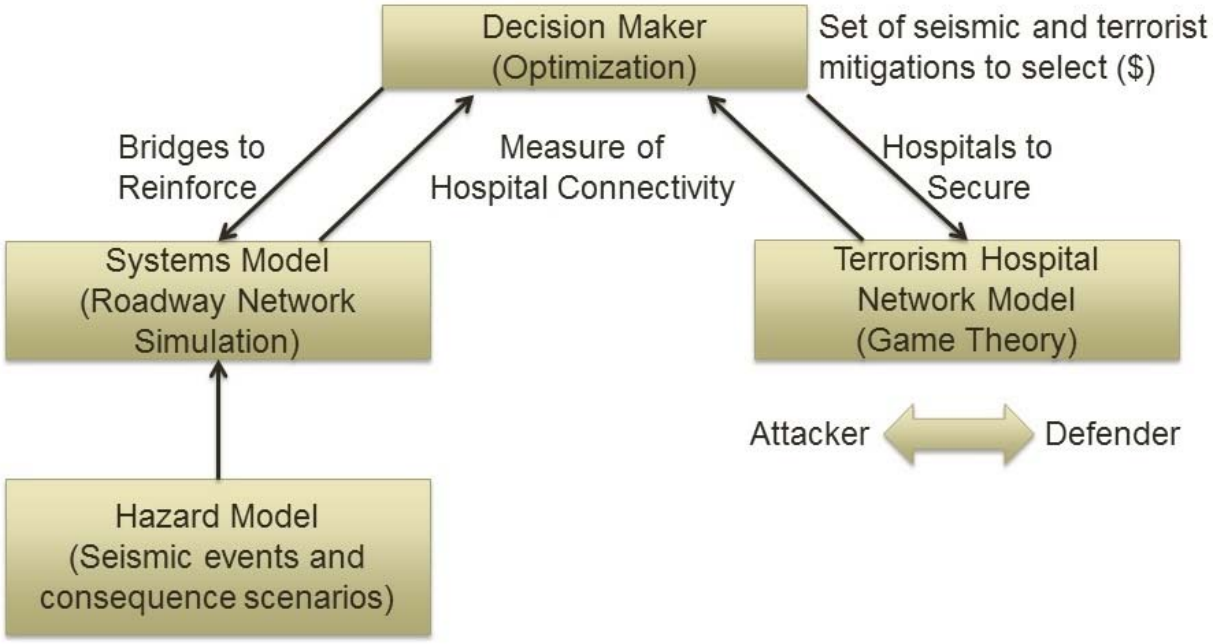


Figure 1: Decision Maker Solution Procedure

In this case study, the decision maker is primarily concerned with ensuring good hospital connectivity for the greatest number of people. To achieve this goal, a single budget must be split between mitigations against both seismic and terrorism threats. For the seismic threat, bridges are selected for reinforcement that will minimize hospital connectivity loss when an earthquake event occurs. For the terrorism threat, hospitals are selected for security upgrades to prevent successful terrorist attacks which would render them unusable. In each case, a DTA evaluation of the post-event roadway network performance is used to calculate the hospital connectivity. The seismic evaluation requires that the roadway network simulation be performed across all consequence scenarios produced by the hazard model. The terrorism performance evaluation requires that a game-theoretic, attacker-defender simulation be run

using the hospital network in conjunction with the roadway network. Since the seismic threat is probabilistic in nature but terrorism is more of a perceived threat, two separate investment planning optimizations must be performed independently. Once a family of mitigation solutions is developed for each threat, the resulting collection can be integrated by enumerating across all possible combinations, which are tied together by overall investment cost. The final collection of solutions can then be used by the decision maker to make a cost-benefit decision based on their priorities (e.g., they may bias towards mitigating the seismic threat with a medium cost if they think it is more probable than a terrorist attack).

3.1. Objective Functions

The decision maker's optimization problem is concerned with three different objectives (described below) for which the goal is to minimize each one. Though not considered in this case study, a fourth objective, Travel Time, is presented, which could be included in the decision-making process. Because maximum infrastructure protection is tied to investment cost, it is necessary to balance system performance against the monetary expenditure. As such, a Pareto Efficiency approach was taken, rather than creating an arbitrarily-weighted combination of the objective values.

3.1.1. Investment Cost

The investment cost is simply the sum of the investments required for 1) mitigating each selected bridge against the seismic threat, and 2) improving the security for each selected hospital in the solution. The cost of mitigating a bridge was set at 10% of the replacement cost. For simplicity, a cost of \$10M for each hospital security investment is assumed, which causes the mitigation of all 20 hospitals (\$200M) to be roughly equal to the cost of mitigating all bridges (\$194M). It would be fairly straightforward to derive a meaningful value that is proportional to the size of the hospital (number of beds); however, we did not have the required data when this study was performed. The final cost objective is computed by summing the cost of the bridge mitigations selected and the cost of the hospital mitigations selected.

$$\min \sum_{i} c_i^B x_i + \sum_{j,k} c_{jk}^H y_{jk}$$

Where,

c_i^B: is the cost of reinforcing bridge *i*

c_{jk}^H: is the cost of implementing mitigation strategy *j* at hospital *k*

x_i: is a binary decision variable that indicates whether bridge *i* is seismically reinforced

y_{jk}: is a binary decision variable that indicates if mitigation strategy *j* is applied to hospital *k*

3.1.2. Travel-Time Objective

The travel-time objective is a measure of the increase in travel time for the general population when a seismic event occurs. Although this metric is not directly hospital-related, it does form part of the basis for determining the seismic hospital connectivity objective. From a story point of view, this is a side objective that could be used by the decision maker to strengthen the argument for making investments in bridges. The argument would be that improving hospital connectivity also benefits the general population in terms of travel time after a seismic event.

The average travel time on a link within a road network is computed using the volume-delay function developed by the Bureau of Public Roads (BPR):

$$T = t_a \left[1 + \alpha \left(\frac{v_a}{c_a}\right)^{\beta}\right] \qquad \text{(Average Travel Time)}$$

Where:

t_a = free flow travel time on link a per unit time (constant)
c_a = capacity of link a per unit time (constant)
v_a = volume of traffic on link a per unit time (variable)
α, β = constant parameters (typically 0.15 and 4, respectively)

To develop a travel-time objective for all traffic that moves across the network within a fixed time horizon, the number of cars delivered within each 15-minute time interval is tracked. This histogram of traffic data can then be used to generate a cumulative distribution curve from which the travel-time objective is calculated. Because traffic can be "stalled" (prevented from reaching its assigned destination), an additional penalty is applied that is equal to three times the maximum travel time experienced by any carload. The figures below illustrate a hypothetical simulation in which the maximum travel time is 165 minutes, which implies a 495 minute penalty for stalled vehicles. The shaded area to the left of the Travel Time CDF is the Travel Time objective.

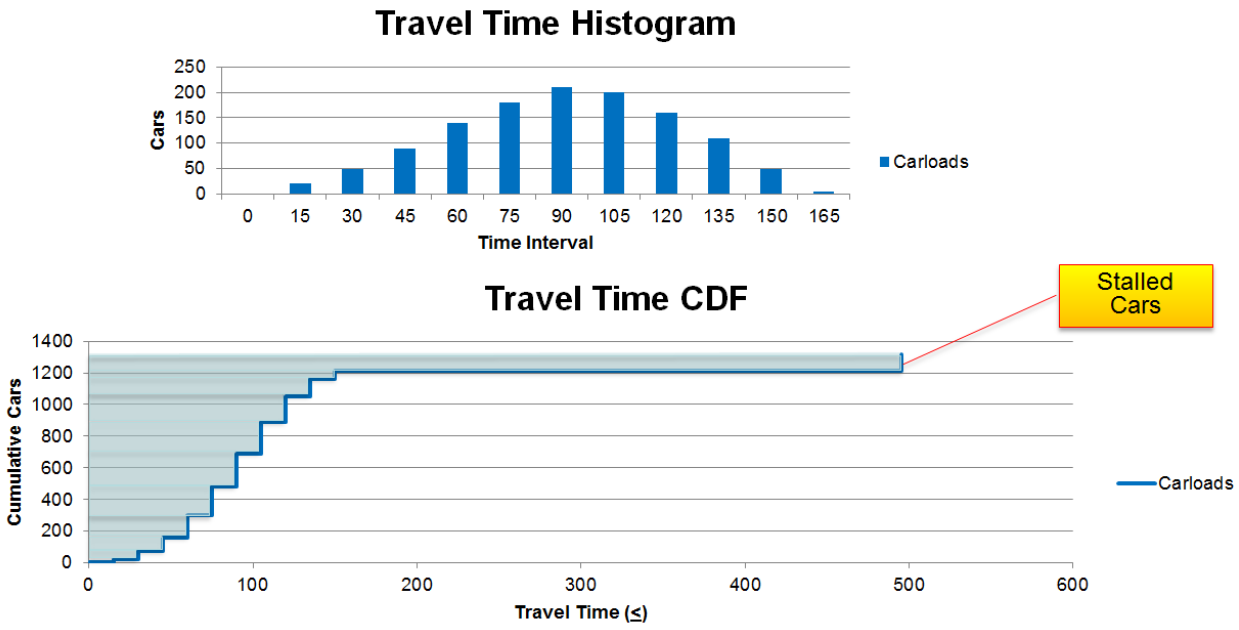


Figure 2: Visual Representation of Travel Time Objective

3.1.3. Hospital Connectivity Objective

Connectivity is a combination of two goals: minimizing the number of people who have no route to any hospital, and minimizing the travel time of those who do have a route to the hospital, with larger importance placed on the first goal. In the Memphis network, there are 1,267 traffic analysis zones (TAZs). Each zone contains a centroid where trips originate and terminate. Connectivity is measured from each zone to the closest hospital. If a zone contains a hospital, the centroid within that zone is identified as a valid destination. When determining hospital connectivity, each origin node is allowed to consider every valid hospital destination. This objective is used to assess the impacts of both the seismic and terrorism hazards.

3.1.3.1. Calculation Procedure

In addition to the highway network parameters used for the DTA, the following inputs are provided:

- A set *W* that contains a weighting factor w_o for each origin node, *o,* in the network.
- A set *H* that contains the destination nodes which are connected to hospitals. *H* is a subset of the destination nodes used in the model.
- A set of travel time breakpoints and a set of penalty terms associated with each breakpoint. This input is used to increase the penalties, because it takes longer to reach a hospital. For example, suppose that a travel time of 0-8 minutes is desired, 8-16 is acceptable but undesirable, and longer than 16 minutes is unacceptable. Assume that a penalty of 0.1, 1, and 10 are applied each bin. This input can be thought of as being

passed into the model in two arrays, [0, 8, 16] and [0.1, 1, 10]. Let *I* represent the number of bins, b_i be the lower break point of each bin *i*, and let p_i be the penalty associated with each bin *i*.

- A large penalty term, *M*, which is applied to traffic that cannot reach a hospital. We chose a value of 10,000 for *M*.

Given these inputs, the connectivity objective can be evaluated using the following steps. This description explains how this objective function is calculated for the seismic hazard. Section 3.1.3.3 describes how this calculation is modified for the terrorism hazard.

1) For each mitigation strategy, run the DTA for each consequence scenario and accumulate all network statistics.
2) For each origin zone and time interval where new traffic is added to the network, determine the shortest path (based on travel time) to any hospital on the network. Use the nodes in *H* to determine which nodes are hospital nodes. Note that the closest hospital could vary during the course of the DTA run due to changes in congestion. Only track these paths for the time periods where traffic is applied to the network which, in our case, is the twelve 15-minute intervals from 6:00-8:45. We ignore the time intervals where traffic is being cleared out of the network. Once this procedure is complete, each origin has a set of numbers that represent the shortest time path to any hospital for the time periods of interest. Note that paths with no connection are also recorded.
3) For each origin zone, we need to measure the quality of its connection to each hospital. Because the network does not change over the duration of the simulation, the results should show that a zone was either connected or disconnected to a hospital for all time periods. For each origin there will be a set of travel times, one for each time interval, so the first task is to convert these values into a single number. Let t_{so} represented how well-connected origin *o* is to any hospital over the window of interest for consequence scenario *s*. In the case where it is not connected, $t_{so} = \infty$; otherwise set t_{so} equal to the 75th percentile of the travel times observed.
4) Finally, the hospital connectivity objective can be calculated. At this point the term t_{so} should have been determined for each origin in each consequence scenario. The hospital connectivity objective is given below:

$$min \sum_{s=1}^{S} \sum_{o=1}^{O} p_s w_o f(t_{so})$$

Where,

p_s is the probability of consequence scenario *s*
w_o is the weight or importance associated with origin zone *o*
$f(t_{so})$ represents the penalty associated with a hospital connectivity of t_{so}

The penalty term for hospital connectivity is calculated as follow:

$$f(t_{so}) = \begin{cases} p_1 t_{so}, & b_1 \le t_{so} < b_2 \\ p_i t_{so}, & b_i \le t_{so} < b_{i+1} \\ p_I t_{so}, & b_I \le t_{so} \\ M, & t_{so} = \infty \end{cases}$$

For this study the penalty terms shown in Table 1 were used, and *M* was set to 10,000. We assume that anything less than eight minutes is a good response time, so the penalty is low. The penalty increases for longer trips so that the model will focus on improving longer trips instead of shorter trips. Assuming that the longest trip to a hospital will be somewhere on the order of an hour (even if all bridges are gone), the maximum penalty will be around 600, which is a much lower penalty than for zones disconnected from hospitals (where the penalty is around 10,000).

Trip Duration (minutes)	Penalty
0-8	0.1
8-16	1
16-24	2
24-32	4
32-40	7
40+	10

Table 1: Hospital Travel Time Penalties

3.1.3.2. Zone Importance Determination

The weight associated with each zone was determined by estimating the number of people in that zone. This estimation procedure is difficult to make accurate, because people move to different zones throughout the day. The census data contains the population of each zone, the number of employees in each zone and the number of students in each zone. It also provides data on the number of households where 0, 1, 2, and 3+ personnel are employed. This information was used to estimate the number of people who are employed in each zone. In the case of 3+ personnel per household, it was assumed that only 3 personnel were employed. The average number of people in a zone over the course of a day was used to determine the weight. The average number of people in each zone over the course of a day was computed as follows:

$$\left(\frac{2}{3}\right)\{zone\ population\} + \left(\frac{1}{3}\right)\{zone\ population - employed\ residents + students + employees\}$$

It was assumed that people will spend 1/3 of their day at school or work. During this part of the day, we assume that employed residents will leave and that students and employees will arrive. If a person is employed in the zone in which they live, they will leave as an employed resident and arrive as an employee. Student residents should be subtracted out of the zone population, but there was no data on the number of students per household. It was assumed

that for the remaining 2/3 of the day, people will reside in their home zone. While this estimation is far from perfect, it should at least provide a reasonable estimate of which areas contain more people. Also, it is less important that each zone be represented exactly, because the bridge investment decisions that the model makes are more likely to affect regions containing many zones, rather than a specific zone. Given this information, as long as larger weights reflect regions where there are more residents, the correct behavior should be seen in aggregate.

Each weight was computed by calculating the average population for its associated zone, then rescaling it to range from zero to one, based on the maximum population. Two corrections were made to the weights. First, the network contains "external" zones that are used to represent traffic flowing in and out of the Memphis area. Because these zones do not represent real geographic areas, it does not make sense to measure how connected they are to hospitals. For these zones, a weight of 0 was used so that they would not impact the optimization. Second, the region containing the Memphis International Airport was assigned an artificial weighting of 1. The calculation above produced a low weight for this zone, because it has a very small population. However, because there are a large number of people in and out of the airport in a day, it was manually given a high priority. Figure 3 shows the importance weight of each zone. Higher weights occur in the more densely populated metropolitan areas. The disconnected triangular regions represent the external zones.

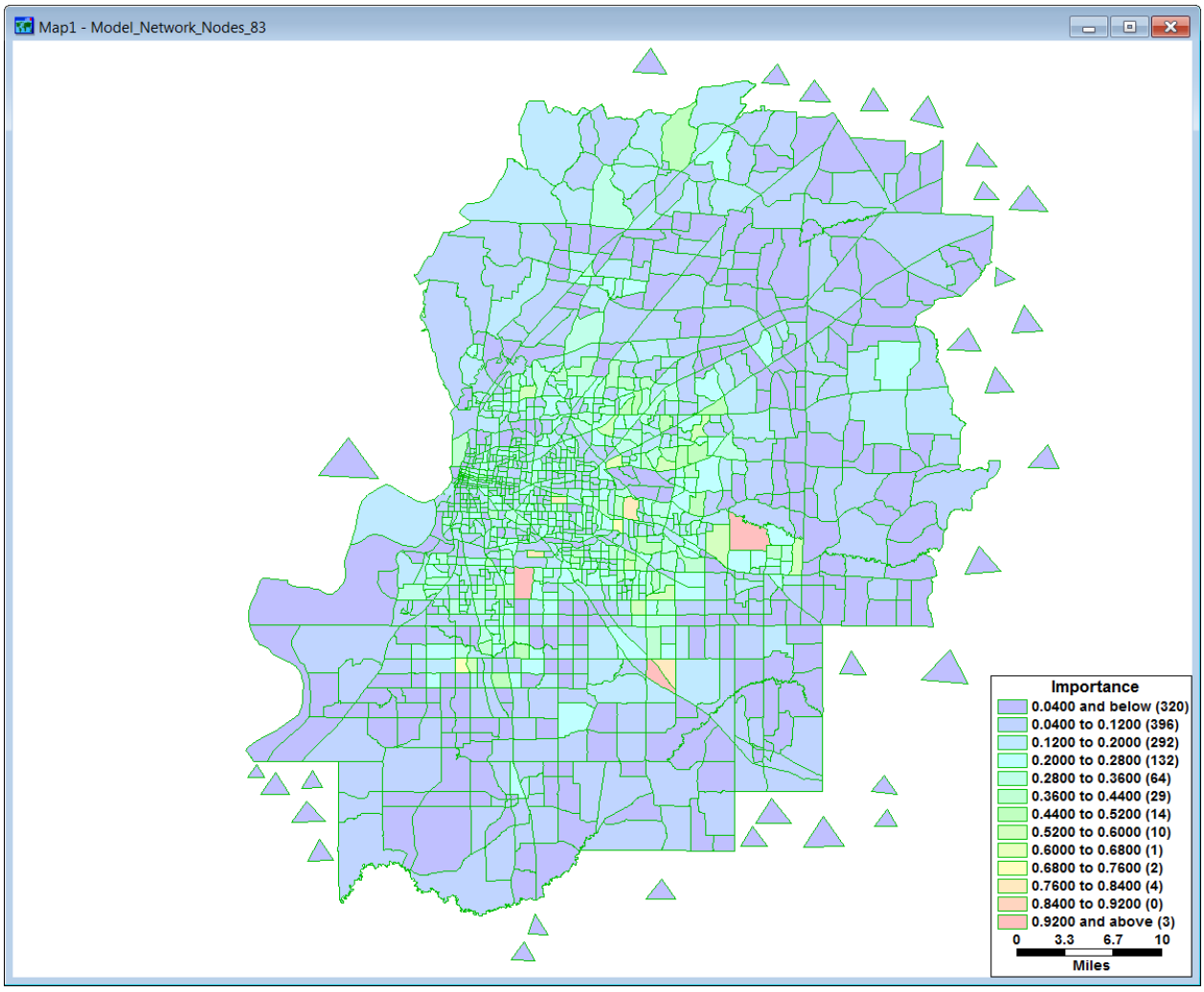


Figure 3: Memphis Importance Zones

3.1.3.3. Differences Between Seismic and Terrorist Objective Calculations

The hospital connectivity objective is calculated in a similar fashion for both the seismic and terrorism hazards; however, there are some small differences. For the seismic hazard, connectivity is affected when bridges are damaged, causing restricted or precluded traffic flow on connected highway links. For the terrorism hazard, hospitals can be completely removed from the network (the terrorist has successfully attacked and disabled them) such that the minimum hospital travel-time calculation only considers the remaining hospitals. As such, a single DTA run can be performed "up front" to calculate travel times to all hospitals from each origin. These values are then stored, and a lookup procedure is used to assess the objective function for each scenario where different hospitals are successfully removed from the network due to a terrorist attack.

3.1.3.4. Other Input Data

A list of the region that contains hospitals is also required. For this study, 20 TAZs were identified that contained hospitals in the Memphis region. Finally, the terrorist's attack budget, in terms of "number of hospitals", is also provided as input.

3.2. Decision Variables

The decision variables for the optimization are broken into two parts: one set for the seismic threat and one for the terrorism threat. For the earthquake threat, each bridge, *i*, has a decision variable, x_i, which takes a value of one or zero indicating if it will be seismically reinforced. For the terrorist threat, a collection of zero or one decision variables y_{jk} are used to indicate whether mitigation strategy, *j*, will be applied to hospital, *k*. Each hospital will have exactly one mitigation strategy for a given solution. For this study, hospitals can either be mitigated (have increased security) or not.

3.3. Solution Procedure

The solution procedure consists of two pieces: computation of the optimal bridge mitigation strategy due to a seismic threat, and computation of the optimal hospital mitigation strategy due to a terrorist threat. This approach is possible because 1) the goal is not to try to find solutions that honor a budget limit, and 2) the hazards and mitigation strategies are independent of each other, so we can consider each hazard separately. This section summarizes how the mitigation solutions for each hazard are generated separately, then presents how they are combined to form one set of solutions on a Pareto frontier.

For the seismic hazard, we use the approach as described in Brown, et al. (2013), which involves a single run of the optimization to determine the Pareto points that consider the total travel time, hospital connectivity, and seismic portion of the cost. For the terrorist hazard, we can execute a separate optimization that determines Pareto points for terrorism mitigation costs and terrorism hospital connectivity objectives. Once this is accomplished, all pairs of solutions from both optimizations are combined to produce a new set of solutions, each of which contains four components: the seismic hospital connectivity objective value, the terrorism hospital connectivity objective value, the seismic travel-time objective, and the total cost (i.e., the terrorism mitigation cost plus the seismic mitigation cost). The final set of Pareto points can be determined from this set of solutions.

3.3.1. Seismic Threat Solution Procedure

For the seismic hazard, we use the approach as described in Brown, et al. (2013), which involves conducting a single run of the optimization to determine the Pareto points that depend on the total travel time, hospital connectivity, and the seismic portion of the cost. The high-level solution procedure can be summarized in the following points and Figure 4.

1. **Identify hazard-consistent earthquake scenarios**: The USGS has cataloged 433 historic events and 20 synthetic events that represent the seismic hazard in the NMSZ.

Using the non-linear optimization process described in Shiraki, et al. (2007), it is possible to reduce this large collection of events to 8 representative earthquakes, of which only 2 pose a significant threat to the bridges in the region.

2. **Identify hazard-consistent consequence scenarios for each earthquake scenario:** The standard method for developing probabilistic consequence scenarios is to use Monte Carlo sampling, however, this requires on the order of hundreds of iterations, which is impractical for performing the optimization we are interested in. As such, we proposed a non-linear optimization procedure, as described in Gearhart, et al. (2013), to create a much smaller number of scenarios (20 per earthquake event) by using FEMA's Hazus loss estimation tool in conjunction with a non-linear optimization.
3. **Assess objectives for post-event system performance:** The DTA is used to the travel time across the network, which also feeds into the computation of hospital connectivity loss. A DTA evaluation is required for each consequence scenario, such that the total travel time and hospital connectivity loss objectives are calculated using a weighted aggregation across all scenarios.
4. **Optimize investment:** The decision regarding which bridges to seismically reinforce drives this optimization. There are three objectives that must be simultaneously minimized for this multi-objective optimization: mitigation cost, travel time, and hospital connectivity loss. The combination of a genetic algorithm (GA) with a local search, run in parallel on the Red Sky supercomputer, is used to subsample the solution space in an intelligent fashion.

The last two steps are repeated multiple times: a GA crossover step is used to combine the solutions at each iteration from each independent process.

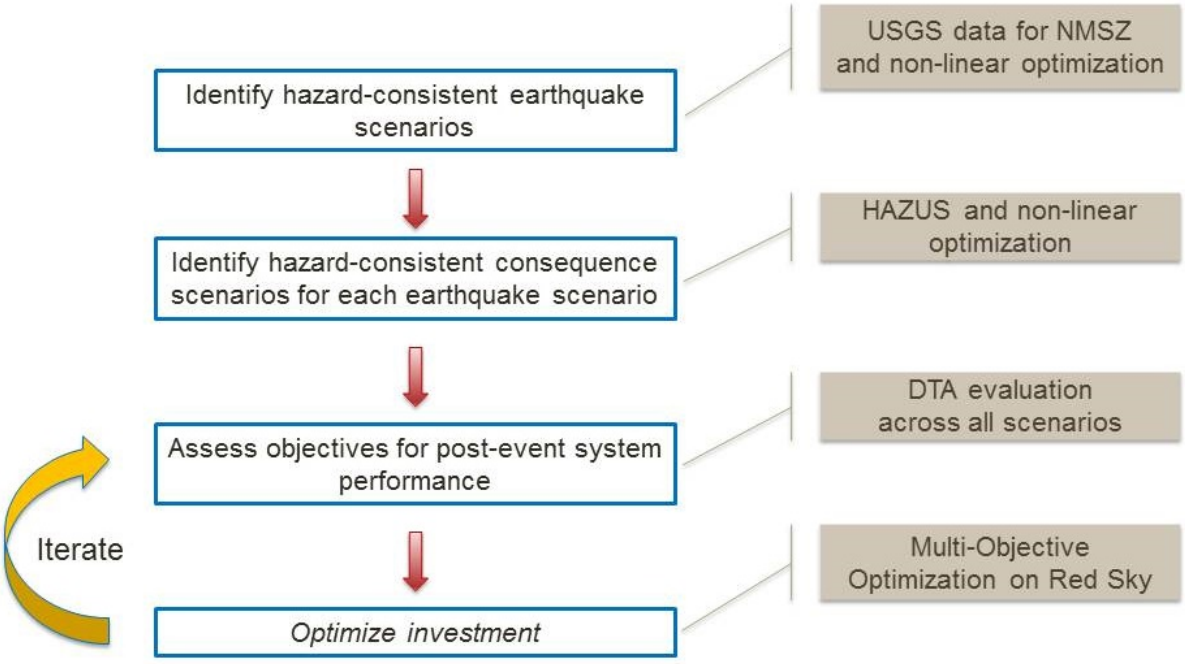


Figure 4: Seismic Investment Optimization

3.3.1.1. Algorithm Summary

A single run of the optimization requires that multiple mitigation strategies be evaluated by measuring the post-event network performance. There are 2 possible earthquake events, each of which has 20 possible consequence scenarios. The DTA requires 11 minutes to evaluate the 40 scenarios required for a single solution. There are 286 bridges in the road network that can be reinforced. Consequently, there are 2^{286} different reinforcement permutations, which is approximately 10^{86} or roughly the number of particles in the universe! Because it is not possible to examine all of the permutations (which would require an approximate runtime of $2x10^{80}$ years), the solution space is sub-sampled by applying a genetic algorithm heuristic and evaluating multiple, random mitigation strategies in parallel, using a collection of compute nodes. At initialization, each node is assigned a random bridge mitigation strategy that indicates the bridges to be reinforced. Twenty consequence scenarios are constructed for each earthquake event based on the bridge mitigation strategies. For a given scenario, each bridge will be in a known damage state ranging from NONE (no impact) to COMPLETE (total loss of use of the affected road link(s)). Network performance is evaluated for each scenario by executing the DTA simulation over a 3-hour time horizon at 15 minute time intervals. Two objective values are produced from the DTA evaluation: travel time across the network, and hospital connectivity loss. The objectives are weighted by the scenario and (normalized) earthquake probabilities, and summed across all scenarios and earthquakes.

To determine the objective contribution based on each earthquake event, the relative weighting based on the event probability is derived. The sum of the two different earthquake event probabilities is not equal to one; however, we are only interested in the fractional contribution of each, so the following normalization is applied:

$$w_x = \frac{p_x}{\sum_{e=1}^{E} p_e}$$

Where:

w_x is the normalized weight contribution due to earthquake event *x*
p_x is the probability of earthquake event *x*
p_e is the probability of earthquake event *e*

Total objective for a single DTA evaluation:

$$C + \sum_{e=1}^{E} \sum_{s=1}^{S} w_e p_s (H_s + T_s)$$

Where:

C is the total cost of the bridge mitigation strategy in dollars
w_e is the weight of earthquake event *e*
p_s is the probability of consequence scenario *s*
H_s is the value of the hospital objective for consequence scenario *s*
T_s is the value of the travel time objective for consequence scenario *s*

A multi-objective optimization is performed using the total objective described above as the basis. Because it is unlikely that all three objectives are *equally* important to the final solution, a Pareto Frontier of solutions is generated. To evaluate each new solution, *n*, its Euclidean distance from the closest point on the current frontier is used as the fitness function.

$$F(C,H,T) = arg\ \min_{x} f(x) \coloneqq \left\{ x \middle| \sqrt{(C_x - C_n)^2 + (H_x - H_n)^2 + (T_x - T_n)^2} \right\}$$

Where:

C_x is the cost of the bridge mitigation strategy for Pareto solution *x*
C_n is the cost of the bridge mitigation strategy for candidate solution *n*
H_x is the value of the hospital objective for Pareto solution *x*
H_n is the value of the hospital objective for candidate solution *n*
T_x is the value of the travel time objective for Pareto solution *x*
T_n is the value of the travel time objective for candidate solution *n*

The sequence of steps for a single optimization run is as follows:

1) The master process spawns the collection of sub processes using MPI (Message Passing Interface) and utilizes RMI (Remote Method Invocation) for inter-process communication and data transfer.
2) An initial run on each node establishes the baseline Pareto frontier by evaluating the two extreme strategies: all bridges reinforced, and no bridges reinforced. This frontier is then used as the fitness function for all local evaluations that start with a random reinforcement strategy across all bridges.
3) A local search procedure (as described in the next section) is performed by evaluating the initial mitigation strategy, along with a small number of strategies that are within the same neighborhood.
4) After a fixed number of iterations, the collection of "best" (closest to the frontier) solutions are sent back to the master process from each sub-process.
5) The master updates its Pareto frontier with the solutions from all "child" processes.
6) A collection of "parent" solutions is created by making a random selection from all new solutions, biased by how close they are to the frontier.
7) New mitigation strategies are formed using genetic crossover and mutation of the parent solution strategies, as described in the pertinent sections that follow.
8) The new strategies are partitioned out to the sub-processes, along with the updated frontier, so that another iteration can be executed.
9) This procedure is repeated a fixed number of times, as each iteration improves the Pareto frontier.

3.3.1.2. Local Search Procedure

The local search solution procedure is used to refine the initial solution on each local compute node. The first iteration computes the full objective value across all 40 consequence scenarios. Each local search iteration is a single bridge-mitigation swap off of the initial mitigation strategy (i.e., changes a single bridge from reinforced to not reinforced or vice versa) and requires a new evaluation across all scenarios. Changing the reinforcement

strategy of a single bridge will likely not change its damage state across all consequence scenarios, hence, only those scenarios in which the bridge damage state changes need to have their objective values reevaluated. The affected scenario objective values from the initial solution are subtracted, and the new scenario objective values are added (making sure to apply the correct weighting based on scenario and earthquake event). The bridge selected for each swap is biased according to the number of scenarios that it changes. For example, a bridge reinforcement change which impacts 30 scenarios is more likely to be selected over one that affects only 20 scenarios. The probability for a bridge being selected for a reinforcement state change (changing to either being reinforced or to not being reinforced) is based on the number of scenarios, as follows:

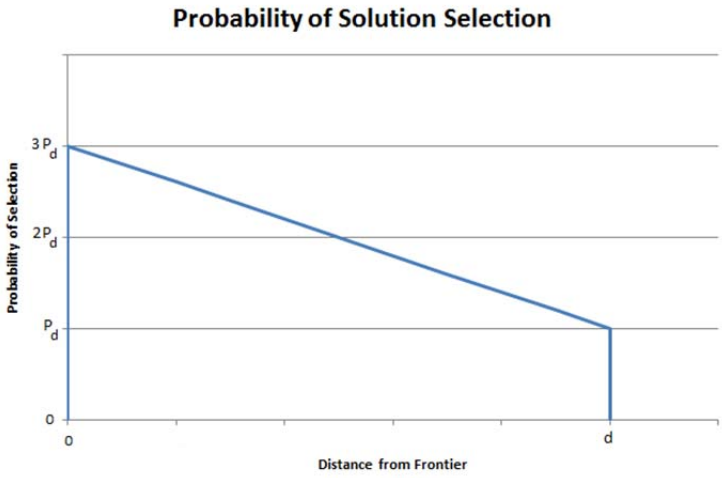
$$P_x = \frac{N_x}{\sum_{b=1}^{B} n_b}$$

Where:

P_x is the probability of selecting bridge x for a reinforcement change
N_x is the number of scenarios impacted by changing the reinforcement of bridge x
n_b is the number of scenarios impacted by changing the reinforcement of bridge b
B is the total number of bridges in the network that can be reinforced

3.3.1.3. Crossover Strategy

The crossover strategy is based on first making a random selection of the new solutions biased towards those that are closest to the frontier. The probability of selecting a solution on the frontier is chosen to be three times as probable as selecting the solution that is furthest from the frontier. A plot of this relationship appears in Figure 5, below. Additionally, an illustration of this biasing method is present in the table of hypothetical data, which shows 10 solution points at various distances from the frontier. In this example, the maximum distance across all solutions is 10, which has a selection probability of 0.05. The total probability across all solutions is 1, as required.



Solution ID	Distance	Probability of Selection
1	0	0.15
2	0	0.15
3	1	0.14
4	2	0.13
5	3	0.12
6	8	0.07
7	7	0.08
8	10	0.05
9	9	0.06
10	10	0.05

Hypothetical Data

Figure 5: Illustration of Biased Selection

Where:

d is the maximum distance from the frontier across all solutions

P_d is the probability of selecting a solution that is the maximum distance from the frontier

Using this biased-selection strategy, the number of solutions selected from the new population is equal to the number of parallel processes being executed. Two new mitigation strategies are created by randomly selecting two strategies from the current solutions, choosing a random cut point, and reassembling the sequences by swapping the two halves. The example below illustrates this process for two mitigation strategies.

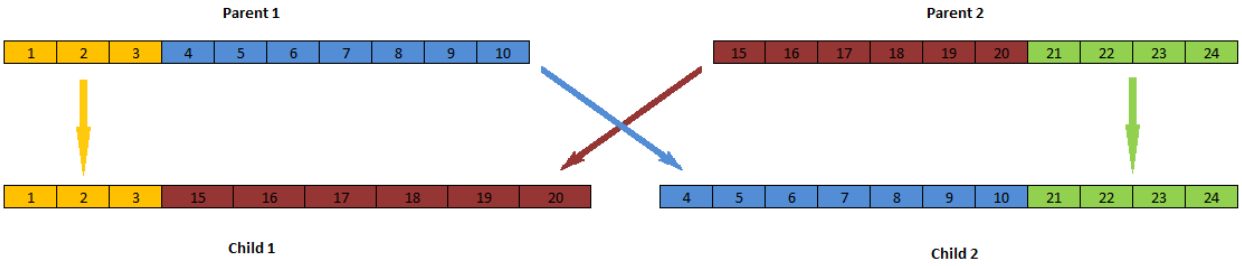


Figure 6: Illustration of Genetic Crossover

3.3.1.4. Mutation Strategy

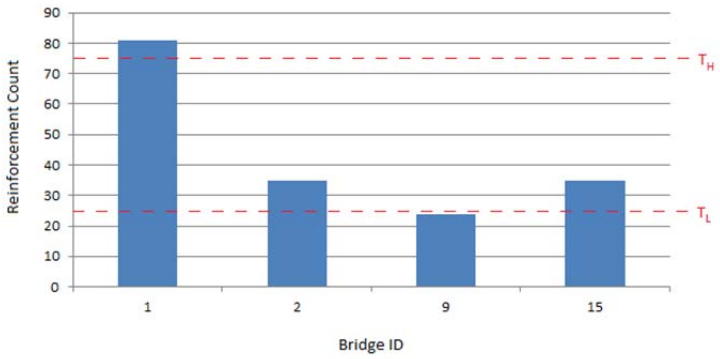
After a number of genetic crossovers, there will be a tendency for the mitigation strategies to converge towards a similar collection of strategies. To create greater diversification in the solution space, mutation is employed based on the similarities between the different mitigation strategies. The *similarity ratio* indicates the similarity across all mitigation strategies in the current solution set based on the number of bridges with the same reinforcement strategy (either reinforced or not). The following steps are used to calculate the similarity ratio and resultant mutation count:

- Accrue the bridge statistics across the current mitigation strategies. This action is essentially counting the number of times each bridge is reinforced.
- Select a threshold, T_H, at which a bridge is considered to be reinforced *most* of the time. Conversely, bridges below $(100 - T_H)$% are considered to be unreinforced *most* of the time.
- Make a count of the number of bridges in "similar" states and calculate a similarity ratio by dividing by the total number of bridges.
- The number of bridge states is calculated as the number of bridges times the number of mitigation strategies.
- The number of bridge state changes that should be changed (mutation count) is then the similarity ratio times the mutation rate times the number of bridge states.

As an example, consider a network with four bridges with IDs 1, 2, 9, and 15. One hundred mitigation strategies are generated with the reinforcement statistics as summarized in the table and histogram in Figure 7. Of the newly generated strategies, ten must be randomly selected and mutated due to the similarity ratio, also shown below.

Reinforcement Counts

Bridge ID	Reinforcement Count
1	81
2	35
9	24
15	35



Mutated Strategies

Original Reinforcement Strategy	Mutated Strategy
1, 2, 9	2, 9
1	1, 9
1	[no bridges]
1, 2, 15	1, 2, 9, 15
1	1, 9
2	2, 9
1, 9	9
1, 2, 15	1, 2, 9, 15
1, 2	1, 2, 9
1, 9	9

Figure 7: Illustration of Genetic Mutation

The high threshold is set at 75%, which indicates that a bridge must be reinforced 75 times out of 100 (or greater) to be considered reinforced *most* of the time and 25 times (or fewer) to be considered unreinforced *most* of the time. From the histogram, it is clear that bridge 1 is reinforced most of the time, whereas bridge 9 is unreinforced most of the time. This situation gives a similarity ratio of 0.5 (i.e., 50% of the bridges have similar reinforcement states across all mitigation strategies). The number of bridge states is 400, which, when combined with a maximum mutation rate of 5%, produces a total mutation count of 20.

3.3.2. Terrorism Threat Solution Procedure

For this study, a game-theoretic approach is taken, using an attacker-defender model to find the optimal hospital reinforcement strategy for minimizing the worst terrorist attack. The terrorist's goal is to disrupt emergency services for the maximum number of people given a limited attack budget. We assume that the terrorist has a maximum attack budget of *k* hospitals out of a total of *n* hospitals in the region, and that all attacks succeed with probability one, which renders the hospital unusable. For example, if the attack budget is selected to be 7 out of the total of 20 hospitals, there are 77,520 different permutations as defined by the 20-choose-7 binomial coefficient:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!};\ \binom{20}{7} = \frac{20!}{7!\,13!} = 77{,}520$$

Additionally, we assume that mitigating a hospital involves making a security investment that is visible to the terrorist (perfect knowledge) and completely deters them from targeting that hospital. Additionally, the defender can reinforce at most *n-1* of the *n* hospitals, since full

reinforcement would create a non-feasible attack solution. The high-level solution procedure can be summarized in the following points and Figure 8.

1. **Select hospitals to reinforce**: The initial collection of solutions will be a completely random selection, which has a size approximately equal to 10% of the full solution space. After the first iteration, genetic crossover will be used to select the next generation of solutions, such that many of the desirable "genes" (hospitals investments) will be preserved. To ensure a complete family of solutions, it is desirable that there be a fairly uniform distribution across the number of hospitals reinforced.
2. **Determine the worst possible terrorist attack given a fixed attack budget:** An MILP solver (CPLEX) is used to determine the worst possible attack for the selected investment, given that the terrorist has a maximum attack budget.
3. **Optimize investment:** The decision regarding which hospitals should have security investments applied drives this optimization. There are two objectives that must be simultaneously minimized for this optimization: security investment cost and hospital connectivity loss.

Genetic crossover is used to create each new generation of potential hospital investments based on preserving desirable (minimum objective) characteristics, but also to allow random mutation to preserve solution diversity. For our study, the terrorist is assumed to have a maximum attack budget of 7 out of the 20 possible hospitals.

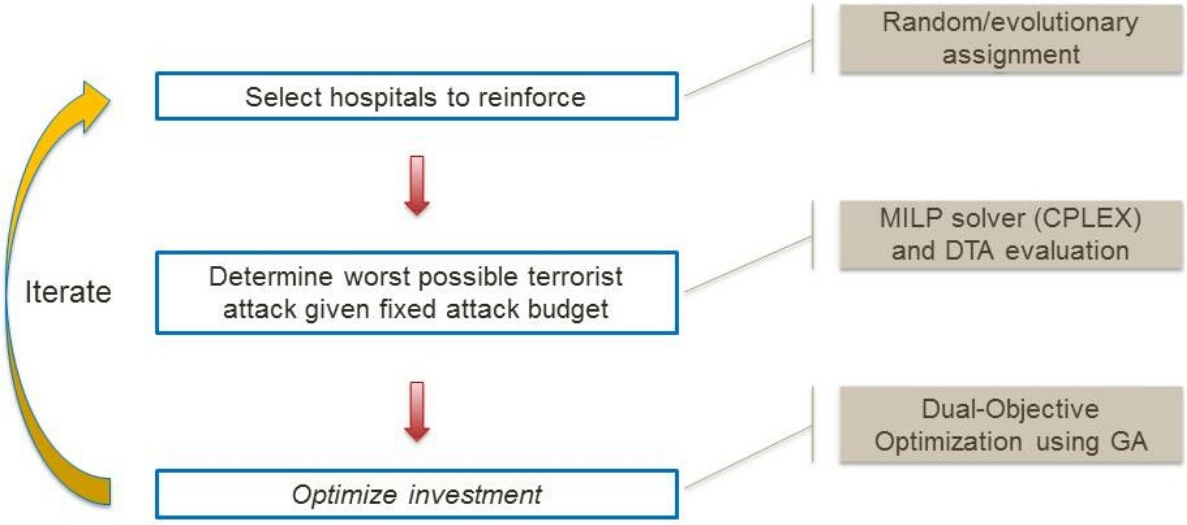


Figure 8: Terrorism Investment Optimization

3.4. Results

For the case study, the seismic and terrorism optimizations are performed independently with the resulting solutions combined by examining all solution permutations. The reason for this separation is that, although there is a probability associated with a new seismic event, there is no calculable probability for a terrorist attack, so the relative weightings would be arbitrary. Decision makers can weight the final trade-off decision based on how significant they perceive the terrorist versus the seismic threat. The Pareto frontier for the seismic optimization (bridge mitigation), considering only bridge mitigation cost and hospital connectivity loss, contains 91 points, as shown in Figure 9.

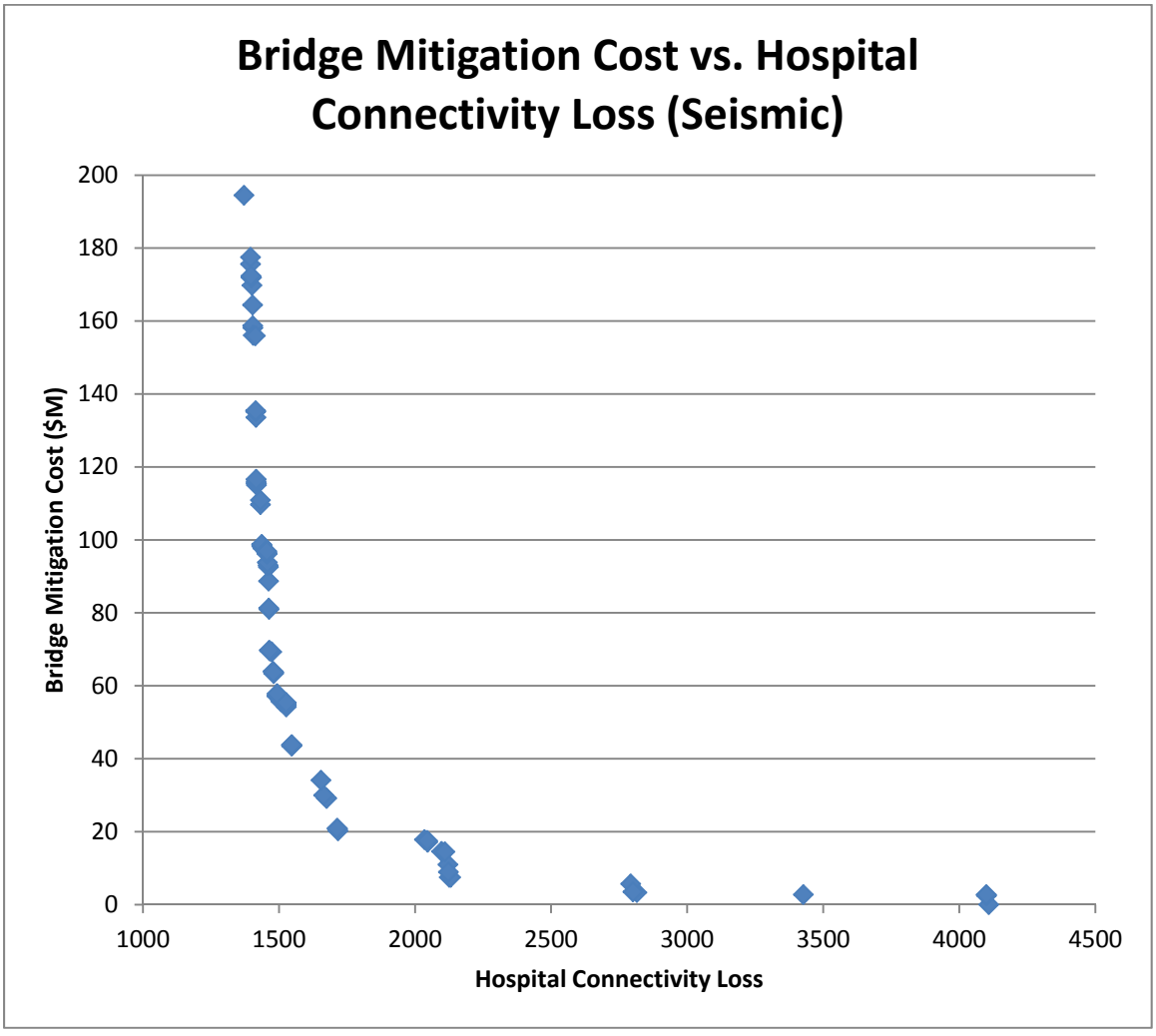


Figure 9: Seismic Investment Pareto Frontier

The Pareto frontier for the terrorism optimization (hospital mitigation), considering only hospital mitigation cost and hospital connectivity loss, contains 15 points, as shown in Figure 10, below.

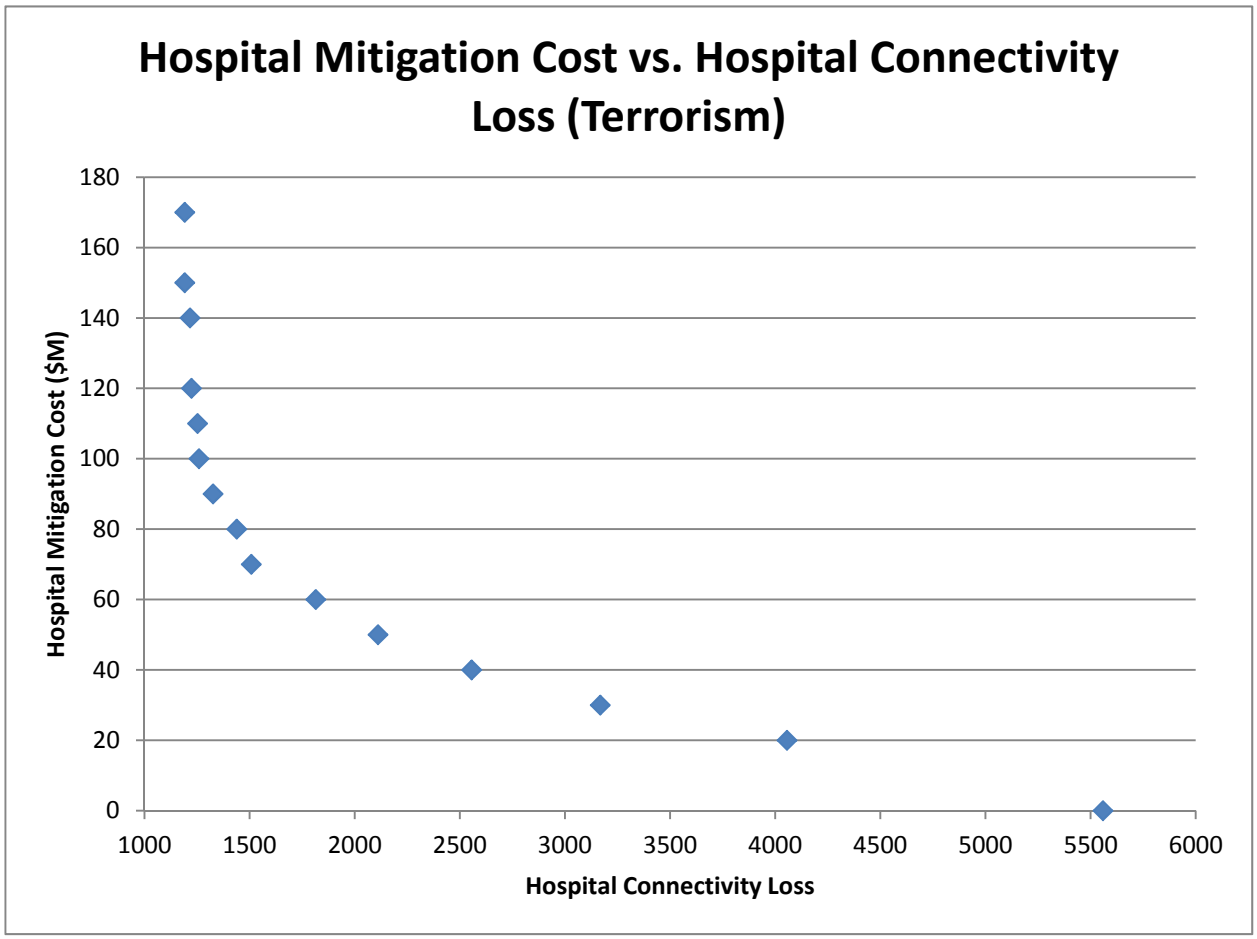


Figure 10: Terrorism Investment Pareto Frontier

The Pareto frontier for the combination of both hazards is the full enumeration of all different solution combinations, which results in 91 * 15 = 1365 Pareto points. The combined frontier uses the combined mitigation cost (hospital plus bridge), the hospital connectivity loss for seismic alone, and the hospital connectivity loss for terrorism alone. Figure 11, below, shows the resulting frontier in 3D along with potential solution choices. In this case, the distribution of points extends into the screen towards the minimum value for each of the three axes. The decision maker can use this surface plot to determine possible solutions in various regions of interest. For example, if there is a significant amount of funding available, the decision can be made to minimize both the seismic and terrorism threats at a high investment cost. Alternately, a medium-cost solution can be selected, which is more focused on seismic mitigation. If funding is highly constrained, then a low-cost solution can be chosen, which balances both seismic and terrorism mitigation, but with much worse resulting performance than the high-cost solution.

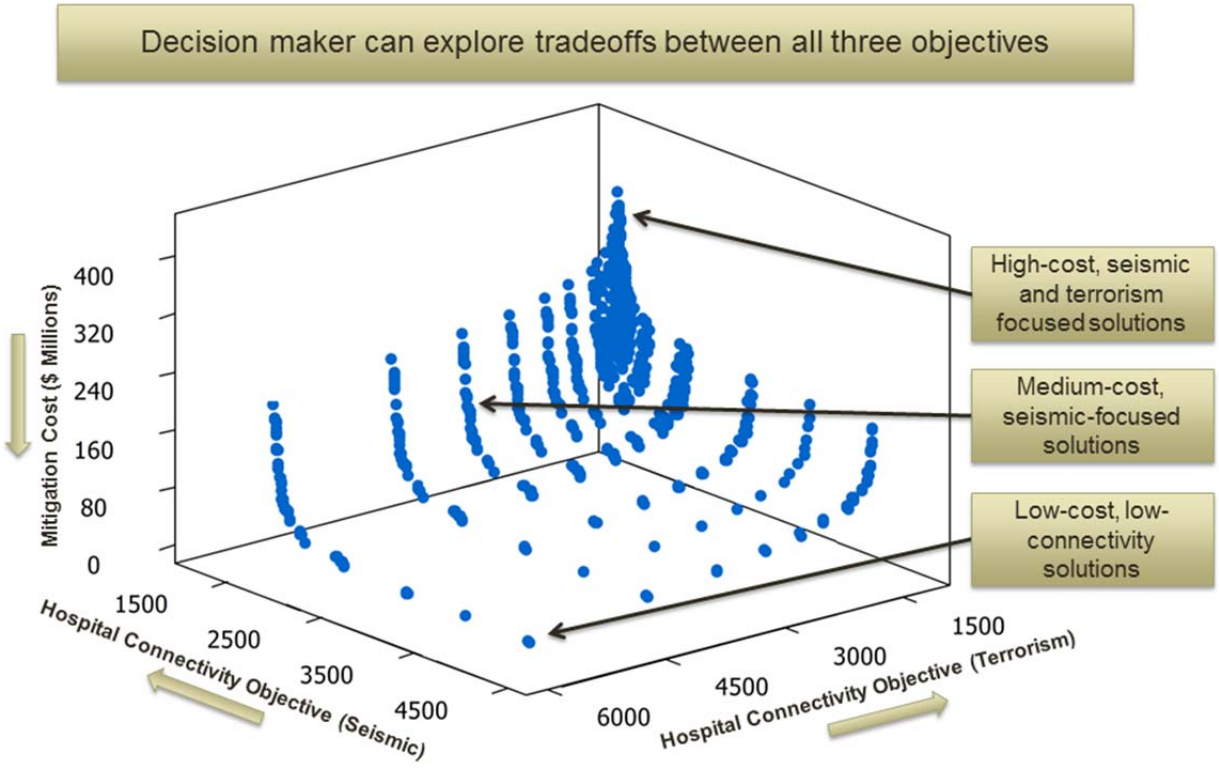


Figure 11: Combined Seismic, Terrorist, and Cost Pareto Frontier (3D)

As an alternative way of viewing the Pareto frontier, but in two dimensions, allows for a somewhat more intuitive presentation. Figure 12 shows the two dimensional representation of the Pareto frontier where the cost is represented by the size of the bubble. One hypothetical line of reasoning could lead the decision maker to filter the results using the following criteria:

1. Remove all solutions that cost more than $100 million.
2. Select solutions where the terrorism hospital connectivity loss is smaller than 2700.
3. Select solutions where the seismic hospital connectivity loss is smaller than 2300.

Figure 13 illustrates the visual partitioning of the solution space and the resultant collection of solution points. Within this smaller collection of solution points, the decision maker could choose the most (circled point in lower left) and least (circled point in upper right) expensive solutions to compare the trade-offs associated with each.

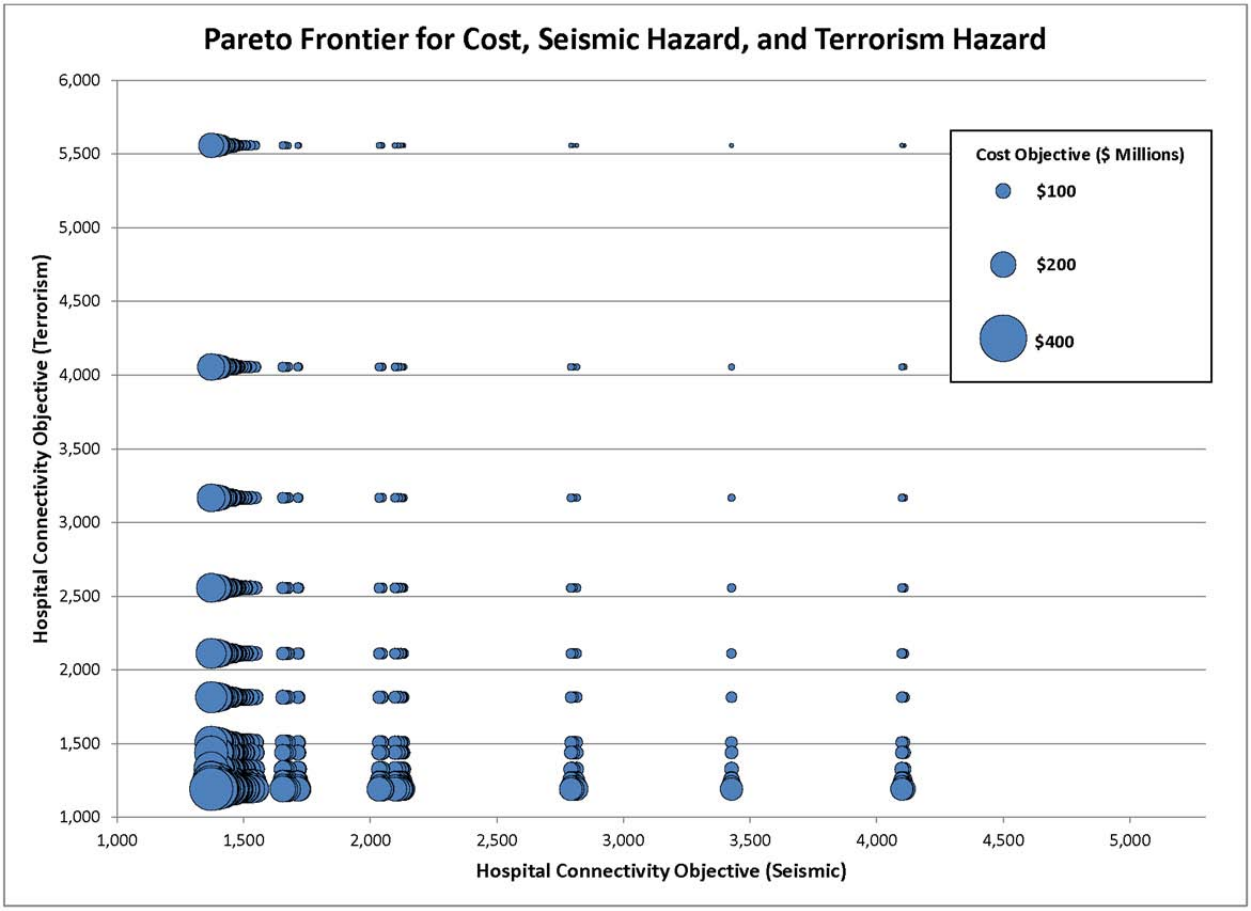


Figure 12: Combined Seismic, Terrorist, and Cost Pareto Frontier (2D)

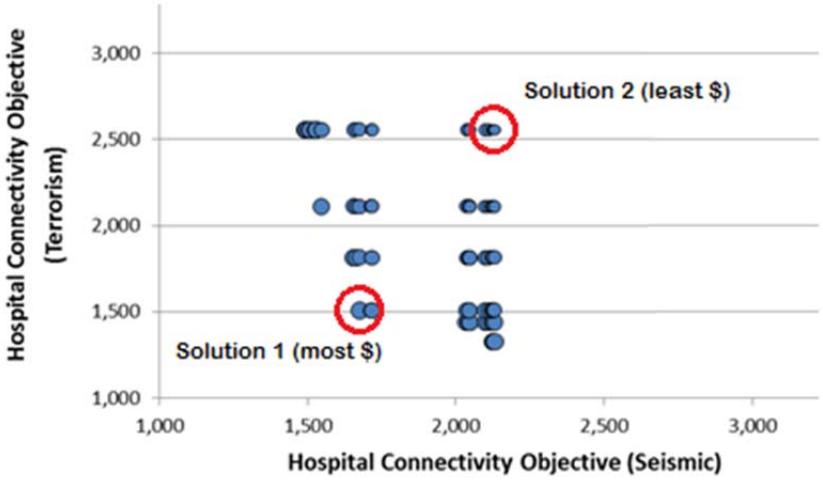


Figure 13: Filtered and Partitioned Solution Space

Table 2 illustrates that by (approximately) doubling the mitigation cost, the seismic objective is decreased from 2131 to 1675; however, it's not immediately clear how this translates to network impact. Table 3 illustrates how these objective values can be mapped to the population that can reach the closest hospital within certain travel times. For both solutions, the pre-event population in the 8-minute time bin is 874,879. After a seismic event, solution one loses 70,416 people out of the 8-minute bin (i.e., they have longer hospital travel times), whereas with solution two, 207,016 people end up with longer travel times. A tool that could provide this type of information to a decision maker would be of great value when making expensive investment decisions that must be defended by sound reasoning.

Table 2: Comparison of Most and Least Expensive Solution Objectives

	Solution 1	Solution 2
Bridges Mitigated	68	19
Hospitals Mitigated	7	4
Cost ($M)	99.2	47.5
Seismic Objective	1675	2131
Terrorism Objective	1508.2	2556.2

Table 3: Mapping from Hospital Objective Values to Population Values

		Solution 1		Solution 2	
Hospital Travel Time (min)	Pre-event Population	Post-Event Seismic Difference	Post-Event Terrorism Difference	Post-Event Seismic Difference	Post-Event Terrorism Difference
0-8	874,849	-70,416	-183,152	-207,016	-422,763
8-16	229,500	+53,021	+173,346	+69,483	+311,641
16-24	73,080	+11,116	+8,387	+79,358	+87,940
24-32	19,809	+5,039	+162	+42,306	+18,746
32-40	3,474	+818	+1,257	+11,738	+4,355
40+	0	+118	0	+3,665	+82
Disconnected	0	+183	0	+346	+0

4. SUMMARY AND CONCLUSIONS

The overarching goal of this LDRD project was achieved by creating an analysis framework which allows investment planning to be performed on multiple, interdependent infrastructures subject to multiple hazards, both natural and terrorism-related. To achieve this goal, we integrated the following capabilities into a rigorous analytic framework: infrastructure modeling, natural hazard analysis, and game theory. As a result, several new models were developed including infrastructure (transportation and electric power), natural hazard (earthquakes) and terrorism. These models were integrated into the investment planning framework using a variety of optimization techniques such as closed form (MILP), heuristic (GA), and non-linear. Numerous artifacts which were produced as a result of this effort, include eight publications in peer-reviewed journals, two technical advances, and a substantial library of software objects which compose the framework.

Next steps in this research could proceed in a variety of different directions. The conversion of roadway networks into more efficient representations utilizing super-centroids could be translated from a manual into a semi-automated process. By augmenting the game-theoretic approach with increased realism in terrorist behavior, a more robust terrorism model could be developed. Advanced visualization and interaction software that would allow a more high-level user to make use of the capabilities for simulating different disaster scenarios and investment strategies would also be of high value. Any of these efforts would augment the ability of the framework to enable decision makers to generate informed cost-benefit decisions.

5. REFERENCES

1. Shiraki, N., M. Shinozuka, J. E. Moore, S. E. Chang, H. Kameda, and S. Tanaka. "System Risk Curves: Probabilistic Performance Scenarios for Highway Networks Subject to Earthquake Damage." *Journal of Infrastructure Systems*, 2007: 13:43-54.
2. Brown, N., J. Gearhart, D. Jones, L. Nozick, N. Romero and N. Xu, 2011, "Optimization of Scenario Construction for Loss Estimation in Lifeline Networks." *Proceedings of the 2011 Winter Simulation Conference*. Edited by S. Jain, R. R. Creasey, J. Himmelspach, K.P. White, and M. Fu.
3. Brown, N., J. Gearhart, D. Jones, L. Nozick, N. Romero, and N. Xu. 2013. "Multi-objective Optimization for Bridge Retrofit to Address Earthquake Hazards." *Proceedings of the 2013 Winter Simulation Conference* (accepted, available after December 2013). Edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl.
4. Gearhart, J., N. Brown, D. Jones, L. Nozick, N. Romero and N. Xu, 2013, "Optimization-based Probabilistic Consequence Scenario Construction for Lifeline Systems." *Earthquake Spectra* (accepted).
5. Li, A., L. Nozick, R. Davidson, N. Brown, D. Jones, and B. Wolshon. 2012. "An Approximate Solution Procedure for Dynamic Traffic Assignment." *Journal of Transportation Engineering*, 139(8):822-832.
6. Romero, N., L. Nozick, I. Dobson, N. Xu, and D. Jones. 2013. "Transmission and Generation Expansion to Mitigate Seismic Risk." *IEEE Transactions in Power Systems* (accepted).
7. Romero, N., L. Nozick, I. Dobson, N. Xu, and D. Jones. 2012. "Seismic Retrofit for Electric Power Systems." *Earthquake Spectra* (accepted).
8. Romero, N., N. Xu, L. Nozick, and D. Jones. 2012. "Investment Planning for Electric Power Systems Under Terrorist Threat." *IEEE Transactions on Power Systems*, 27(1):108-116.
9. Xu, N., N. Romero, L. K. Nozick, D. A. Jones, and N. Brown. 2013. "A Decomposition Heuristic for a Power Network Interdiction Problem." *IEEE Transactions on Power Systems* (submitted).

APPENDIX A: ROADWAY NETWORK DATA FILES

The data required for a full description of each network is composed of multiple files, each with a different function. The following sections describe the contents of each file.

Highway Network File

The network file contains nodes on the network, the links that join these nodes, and the link attributes. This file is a CSV file associated with configuration property *highway network file*. The following columns are present in the file, which can be either comma or tab delimited (one or the other, but not a mix of both).

- Link ID – The integer ID, which uniquely identifies the link.
- Source ID – The integer ID, which uniquely identifies the link's origin node.
- Destination ID – The integer ID, which uniquely identifies the link's destination node.
- Direction – The direction of the link. A +1 indicates traffic flowing from source node to destination node. A -1 indicates traffic flowing in the reverse direction. This column is necessary, because different directions of a link may have different attributes (e.g., capacity).
- Time – The time in minutes required to traverse the link, under free-flow conditions.
- Capacity – The capacity of the link in carloads per hour.
- Alpha – Alpha value in BPR volume-delay function (can be omitted)
- Beta – Beta value in BPR volume-delay function (can be omitted)

Notes:

- All link and node IDs must be positive integers. This restriction is required so that the negative of the link ID is used internally in the model to identify the reverse direction link.
- The alpha and beta values can be omitted from the file if desired, in which case the configuration file values will be used. The definition of the BPR volume-delay formula is specified in the section on the Travel Time Objective.
- For the Memphis network, there are 2 versions of this file: HwyNet.txt specifies 25,970 links for the full network, whereas merged_HwyNet.txt specifies 12,347 super-links (combinations of multiple standard links). The merged_HwyNet.txt version was created by converting the nodes to super-centroids (collections of nodes), resulting in a network that is 47.5% of the size of the original network.

OD Matrix File

The OD (origin-destination) matrix file specifies all origin and destination nodes, as well as the associated traffic flow that moves between these nodes over the time horizon. The file is a CSV with 3 columns: origin node ID, destination node ID, and traffic flow in carloads per hour. The default filename is ODMatrix.txt and is associated with configuration property *OD matrix file*. The OD matrix is converted to a temporal demand OD matrix when combined with the *timing percentage file*, which defines the fraction of total demand for each time interval.

Timing Percentage File

This file specifies the fraction of the total OD demand at each time interval (typically every 15 minutes). The file is a CSV with a single column (the fraction of demand), and has a row entry for each time interval. The configuration property intervals *per hour* must be set properly to coincide with the entries in this file (e.g., a value of "4" indicates 15-minute time intervals). The default filename is "DepartTiming.txt", and is associated with configuration property *timing percentage file*. This file is used to convert the OD matrix to a temporal demand per time interval.

Bridge Cost File

This file specifies all bridge IDs and their associated mitigation cost in millions of dollars (roughly 10% of the replacement cost). The file is a CSV with 2 columns: ID and cost. The default filename is AllowableCost.txt and is associated with the configuration property *bridge strategy file*.

Bridge-Link Relationship File

This file specifies the different bridge-link relationships for each bridge. It is a CSV file with the following columns: bridge ID, link ID, link direction (1 or -1), and relationship ID (1 = link runs on top of the bridge, or 2 = link runs under the bridge). This file is associated with configuration property *bridge-link relationship file,* with a default filename of BridgeLinkRelation.txt for the original network and merged_BridgeLinkRelation.txt for the super centroid network.

Recovery Plan File

This file specifies a generic recovery plan over time for bridges damaged by an earthquake event, and indicates the fraction of available capacity for links affected by a bridge. It is a CSV file with the following columns:

- Bridge damage state – The damage state of the bridge after the earthquake event ranging from 1 (no damage) to 5 (complete damage).
- Bridge-link relation – The relationship for which the recovery plan applies, which is either 1 (link is on top of the bridge) or 2 (link runs under the bridge).
- Day – The day in the recovery plan starting at 1 (the day after the event).
- Capacity – The fraction of available capacity on links affected by a bridge in the given damage state. This value will vary from 0 (no available capacity) up to 1 (full capacity).

The Recovery Plan file was intended to be used for the change in link capacity over time; however, it is currently only used to determine the link capacity immediately after an earthquake event. This file is associated with configuration property *recovery file*, and has a default filename of RecoveryPlan.txt.

Earthquake File

This file specifies the earthquakes to be used in the optimization. It is a CSV file with two columns: earthquake ID, and probability of occurrence (value between 0 and 1). This file is

associated with configuration property *earthquake file*, with a default filename of Earthquake.txt.

Bridge Damage Probability File

This file specifies the probability of each damage state for all bridges and reinforcement strategies. It is a CSV file with the following columns: earthquake ID, bridge ID, strategy ID, damage ID (value ranging from 1 [no damage] up to 5 [complete damage]), and probability (decimal value between 0 and 1). This file is only required when doing Monte Carlo sampling, and has a default filename of BridgeDamageProb.txt associated with configuration property *bridge damage probability file*. When Monte Carlo sampling is not used, a mitigated scenarios file must be present.

Mitigation Scenario File

This file specifies a collection of static scenarios in a CSV file format, has a default filename of MitigatedScenarios.txt, and is associated with configuration property *mitigation scenario file*. The first row of this file contains the column names: Earthquake, Scenario, Probability, 1, 2, …, N. The interpretation of the columns is as follows: [quake ID],[scenario ID],[scenario probability/weight],[CSV of bridge damage states for all bridge IDs 1 through N]. The first entry for any scenario will be assumed to hold the unmitigated damage states. The second entry for an already existing scenario will be assumed to hold the mitigated damage states. The weight is the probability of the scenario given that the associated earthquake occurs. To get the complete, conditional probability weight for the scenario, it must be scaled based on the probability of the associated earthquake. The scaling due to earthquake probability is a multi-step procedure:

- Determine the scaling to be applied to each quake probability such that the sum of the quake probabilities is 1: $1/(P1 + P2 + ... + PN)$
- The full scenario weight is then: (probability of scenario) * (probability of quake) * (quake probability normalization)

Only those bridges for which there are IDs should have their damage states updated. For the Memphis dataset there are 335 bridges of which 14 which are not connected to the network and are omitted. There must be the same number of scenarios for the mitigated case as for the unmitigated case with the same probabilities. For the Memphis dataset there are 20 core scenarios. The file is structured so that we have scenarios 1 – 20 for an earthquake ID that represent no mitigation damage probabilities, then another 20 scenarios with the same quake and scenario IDs that represent mitigation damage probabilities. The next collection of 20 scenarios should be numbered 21 – 40 for the next earthquake ID, etc.

Origin Importance File

This file specifies the importance of each TAZ (Transportation Analysis Zone) based on the population in that zone. It is used for determining the hospital connectivity objective, as described in the section of the same name under the case study (see Section 3.1.3). It is a CSV file with the following columns: EP_ID (endpoint ID: the ID of the centroid node identifying the TAZ), weight (importance of TAZ), AVG_POP (average population in TAZ). This file is only required when doing an optimization around a hospital network, and has the

default filename of OriginImportance.csv associated with configuration property *origin importance file*.

APPENDIX B: AIR SYSTEM MODEL DETAILS

Overview

There are two main components to the model. The first is a large database containing publicly available airline data obtained from the Bureau of Transportation Statistics website. The second is an optimization model implemented in Optimization Programming Language OPL. Through a series of database queries, the historical data is manipulated into a manageable dataset that can then be processed by OPL. This manageable dataset contains an estimation of passenger demands for travel between all origin-destination pairs for each airline, as well as a flight schedule (within a specified time window). The goal of the OPL model is either to route as many passengers as possible through the system within the time window, or to minimize the number of "shed passengers" – passengers that are unable to reach their destination.

The queries that create the manageable dataset are dependent on a set of user-defined parameters. Adjusting these parameters allows the user to create different datasets. The datasets may differ by how many airports are represented in the model, which (and how many) flights are included during the time window, how coarsely the time window is discretized, and how many different "types" of passengers there are (a passenger "type" is a group of passengers with a unique triple [i.e., origin, destination, and carrier]).

Before solving the OPL model, the user must determine which of these datasets to use, as well as which of the airports are disrupted. Once these decisions are made, and a few other parameters are specified, the OPL model is solved, resulting in the optimal passenger flow. This flow can then be analyzed to determine the impact that the disruption had on the passenger air system. A pictorial representation of this framework is shown in **Error! Reference source not found.**4.

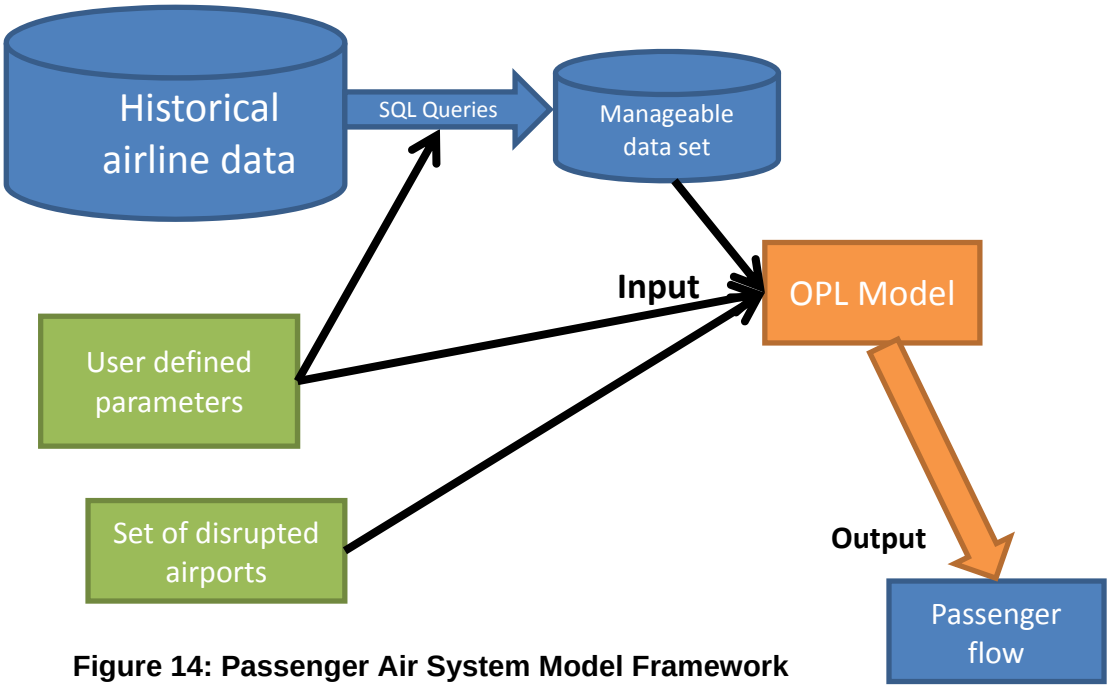


Figure 14: Passenger Air System Model Framework

Optimization Model and Implementation

To model the passenger air transportation system, we use a multi-commodity flow model with side constraints. The multi-commodity flow problem is a network flow problem involving multiple commodities, each with a source node, sink node, and flow demand that must be sent between the two nodes. The goal of the optimization problem is to minimize the cost of sending all of the commodities between their source and sink nodes, while adhering to the "arc" capacities.

In our model, a "commodity" is defined as a set of passengers with the same origin, destination, and airline carrier. The network contains a node for every airport-time period pair. There are three types of arcs in the network. The first type of arc represents actual flights in the air transportation system. The second type of arc connects two nodes representing the same airport in consecutive time periods. The third type of arc is a "dummy arc", which accommodates passengers that are unable to make it from origin to destination by traveling through the air system. Complicating the model are side constraints – in this case, the side constraints are restrictions that are placed on which types of passengers can use which flights (to be discussed in more depth later).

Model Assumptions

Various assumptions are made in the development of this model. First, instead of determining the specific itinerary (or route) a passenger will follow in traveling from origin to destination, only the origin and destination are specified. A model in which actual passenger routes are specified would most likely have been less tractable.

Second, instead of introducing passengers into the system at different times throughout the day, they all enter the system during the first time period. The lack of historical passenger data at the hourly (or even daily or monthly) level makes estimating a distribution for the times that passengers enter the system very difficult. Therefore, for our analysis, it is sufficient to allow all passengers to enter the system during the first time period, effectively allowing the optimization model to determine what time the passengers depart on their first flight of the day.

We also assume that, in the event of disruptions, the flight schedule does not change. In other words, the arrival and departure times of all flights remain the same, and flights that originate or depart from interrupted airports simply do not operate.

Last, instead of solving an integer program (IP), we solve a linear program (LP). While this results in unrealistic (fractional) passenger flows, the added value of an integer solution is not worth the extra computational burden induced by solving an IP.

Notation and Formulation

A formal representation of the model is developed using the following notation:

T : set of time units, indexed by t
A : set of airports, indexed by a

F : set of flights, indexed by f
O_f : origin of flight f
D_f : destination of flight f
C_f : carrier of flight f
t_f^O: time period flight f leaves its origin
t_f^D: time period flight f arrives at its destination
u_f : capacity of flight f
L_a: minimum layover time at airport a (currently $L_a = 0 \quad \forall a$)
K : set of commodities, indexed by k (a commodity is a set of passengers defined by a unique triple: (origin, destination, carrier))
n^k : # of passengers for commodity k
o^k: origin of commodity k
d^k: destination of commodity k
A^k: the set of airports that commodity k is allowed to visit
F^k: the set of flights that commodity k can use

Given this notation, a network is constructed with a node for every airport-time period pair: $(a,t) \in A \times T$. Given an airport, $a \in A$, and time period, $t \in T \backslash |T|$, an arc is added between nodes (a,t) and $(a,t+1)$. For each flight, $f \in F$, an arc is added between nodes (a,t) and $(\bar{a},\bar{t})$ where $a = O_f$, $t = t_f^O$, $\bar{a} = D_f$, and $\bar{t} = t_f^D + L_{\bar{a}}$. A dummy arc is added for each commodity, $k \in K$, connecting nodes $(o^k, 1)$ and $(d^k, |T|)$. This representation can be viewed as a series of chains, one for each airport, that is interconnected by flight arcs and dummy arcs (see Figure 2). The objective is to route the passengers in a way that minimizes the total number of passengers that require dummy arcs to get from their origin node to their destination node (e.g., for passengers of commodity type k, from node $(o^k, 1)$ to node $(d^k, |T|)$). In the example below, the green arcs represent flights, the orange arcs allow passengers to remain at an airport from one time period to the next, and the purple arc is the dummy arc for a group of passengers flying from ABE to ABI with a specific carrier.

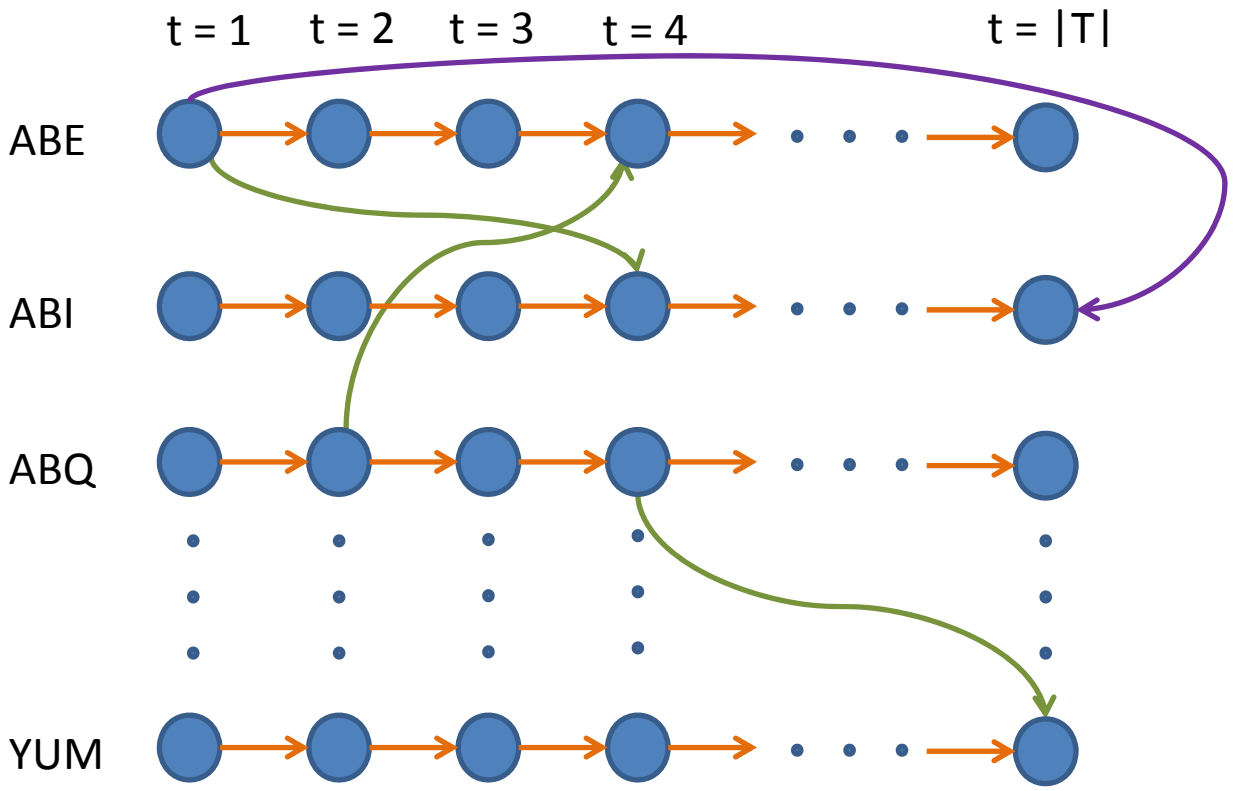


Figure 15: Network Model Representation

The capacities of the arcs corresponding to flights are the capacities of those flights (flights with identical origin, destination, carrier, departure time, and arrival time are combined). All other arcs are uncapacitated. The dummy arcs have a cost of 1 (representing one passenger that cannot make it to its destination), while all other arcs have 0 cost. There are three sets of decision variables:

$x^k_{(a,t)}$: the amount of commodity k (# of passengers) that move from node (a,t) to node $(a,t+1)$

y^k_f: the amount of commodity k that uses flight $f \in F$

z^k: the amount of commodity k using its corresponding dummy arc

The formulation is as follows:

$$\min \sum_{k \in K} z^k$$

such that

$$z^k + x^k_{(o^k,1)} + \sum_{\left\{f \in F^k \middle| O_f = o^k, t^O_f = 1\right\}} y^k_f = n^k \qquad \forall k$$

$$x^k_{(a,1)} + \sum_{\left\{f \in F^k \middle| O_f = a\,, t^O_f = 1\right\}} y^k_f \quad = 0 \qquad \forall k \in K, a \in A^k \backslash o^k$$

$$x^k_{(a,t-1)} + \sum_{\left\{f \in F^k \middle| D_f = a, t^D_f + L_a = t\right\}} y^k_f = x^k_{(a,t)} + \sum_{\left\{f \in F^k \middle| O_f = a, t^O_f = t\right\}} y^k_f \qquad \forall k \in K, a \in A^k, 1 < t < |T|$$

$$x^k_{(a,|T|-1)} + \sum_{\left\{f \in F^k \middle| D_f = a, t^D_f = |T|\right\}} y^k_f = 0 \qquad \forall k \in K, a \in A^k \backslash d^k$$

$$z^k + x^k_{(d^k,|T|-1)} + \sum_{\left\{f \in F^k \middle| D_f = d^k, t^D_f = |T|\right\}} y^k_f = n^k \qquad \forall k$$

$$\sum_{\{k | f \in F^k\}} y^k_f \leq u_f \qquad \forall f \in F$$

$$x, y, z \geq 0$$

The objective function minimizes the number of passengers that travel on dummy arcs. The first set of constraints ensures that n^k passengers leave the node representing commodity k's origin airport at time period 1. The second set of constraints ensures that no passengers of commodity type k originate at any airport other the origin. The third set of constraints ensures that the same number of passengers for each commodity type enter and leave each intermediate node. The fourth set of constraints ensures that 0 passengers of commodity type k terminate at any airport other than the destination. The fifth set of constraints ensures that exactly that n^k passengers reach commodity k's destination airport by the last time period (this set of constraints is actually redundant, and is not included in the OPL model). The sixth set of constraints ensures that no flight's capacity is exceeded, and the seventh set of constraints contains the nonnegativity restrictions.

The number of constraints for the model is $O(|F| + |T| \sum_k |A^k|)$, and the number of variables is $(\sum_k |A^k| + |F^k|)$. Because of this, we use sets $A^k \subset A$ and $F^k \subset F$. This may result in a slightly worse solution (because of the reduced solution space), but the model would most likely be intractable if we allowed every passenger to visit any airport and use every flight. Also, by constructing these sets, we restrict the model from choosing strange routes, such as Albuquerque-LA-Honolulu-San Francisco-Atlanta for a passenger to get from Albuquerque to Atlanta.

OPL Implementation

The implementation of the mathematical model is accomplished using OPL. The project consists of three files: a data file, model file, and a settings file. Two steps must be completed before the OPL model is solved. First, the user must specify the values for a set of six parameters in the `Input_OPL_Parameters` table of the **`AirlineDataLDRD`** database. Second,

the user must run the *Pre_OPL_Run.sql* query and then identify which airports are interrupted in the `Input_Airports_With_Status` table that is generated from the query. Once the user has completed these steps, the OPL code can be run. When the OPL code is run, a unique integer, *`ModelRunID`*, is generated and the following steps are completed:

1) The data and parameters are read in from the **`AirlineDataLDRD`** database.
2) The "uninterrupted" version of the model is solved – the number of shed passengers is minimized when no airports are disrupted.
3) All shed passengers are removed from the system. (Ideally, the uninterrupted version wouldn't have any shed passengers, but because the OD table is an estimation, there will most likely be some unaccommodated passengers).
4) **[OPTIONAL]** The passenger flow is fixed for the commodities that have no passengers flying through the disrupted airports.
5) The "interrupted" version of the model is solved – the number of "shed" passengers is minimized when the specified subset of the airports is interrupted.
6) **[OPTIONAL]** A third LP (linear program) is solved in which a secondary objective is optimized. A constraint is added so that the total number of shed passengers is no more than the optimal number of shed passengers. The secondary objective may be either to obtain a solution with the smallest deviation from the historical load factors, or to find an alternative optimal solution in which the number of passengers that board each flight is below 25% capacity is summed and minimized.
7) The output is exported to the database.

Each *`ModelRunID`* refers to an OPL run in which a *`DatasetID`* and set of inoperable airports have been selected (as well as the values for the other 5 parameters in the `Input_OPL_Parameters` table). When incorporated into other models, it is most likely that the same *`DatasetID`* and OPL parameters would be used for a given analysis, but with different sets of inoperable airports. To automate this process, new model runs can be created from a common dataset and only the `Input_Airports_With_Status` table would need to be modified. This approach allows the model to be solved multiple times in a loop.

Input Parameters

Below are the six parameters that must be specified in the `Input_OPL_Parameters` table of the **`AirlineDataLDRD`** database before OPL is run (this table should only have one row):

- *`DatasetID`*: identifies which dataset will be used by OPL. (The set of airports, carriers, commodities, and flights is different for each dataset. The datasets have been constructed by running SQL queries on the historical airline data.)
- *`FixUninterruptedCommodities`*: If equal to 1, then, after the uninterrupted model is solved, all passengers for any commodity that doesn't have any passengers flying through interrupted airports have their routes fixed.
- *`UninterruptedSharing`*: If equal to 1, then, in the uninterrupted model, passengers flying with a specific carrier may be able to use flights from other carriers (more on this later). If equal to 0, then passengers may only use flights provided by the carrier they purchased their tickets from.

- *InterruptedSharing*: If equal to 1, then in the interrupted model, passengers flying with a specific carrier may be able to use flights from other carriers. If equal to 0, then passengers may only use flights provided by the carrier they purchased their tickets from.
- *MinimizeEmptyFlights*: If equal to 1, after solving the interrupted LP, a third LP is solved that attempts to find an alternative optimal solution, in which, for each flight below 50% capacity, the number of passengers is summed and minimized.
- *MinimizeLoadFactorDeviation*: If equal to 1, after solving the interrupted LP, a third LP is solved that attempts to find an alternative optimal solution, in which the load factor deviation is minimized.

Also, the set of interrupted airports must be identified by modifying the *Interrupted* column of the Input_Airports_With_Status table (set the value equal to 1 if the airport is interrupted; otherwise, set the value to 0 and designate the time periods that the airport is interrupted).

Note: At most one of the parameters: *MinimizeEmptyFlights* and *MinimizeLoadFactorDeviation*, may be set to 1.

Input Data Tables

Below is a list of the data tables used as input by the OPL model:

- Input_OPL_Parameters: This table contains the *DatasetID* for the dataset to be used, as well as the values for the following OPL parameters: *FixUninterruptedCommodities*, *UninterruptedSharing*, *InterruptedSharing*, *MinimizeEmptyFlights,* and *MinimizeLoadFactorDeviation*.
- Input_Dataset_Input_Parameters: Given the *DatasetID* value from Input_OPL_Parameters, OPL pulls from this table the number of time periods per hour, as well as the start and end of the time window.
- Input_Airports_With_Status: This table contains the set of airports to be used for the model, as well as an indicator for each airport identifying whether it is disrupted or not.
- Input_Commodities: Given the *DatasetID* value from Input_OPL_Parameters, OPL pulls a list of commodities from this table. Each commodity has an origin, destination, carrier, number of passengers, and a *CommodityID*.
- Input_Flights: Given the *DatasetID* value from Input_OPL_Parameters, OPL pulls a list of flights from this table. Each flight has an origin, destination, carrier, capacity, departure time period, arrival time period, and a *FlightID*.
- Input_Commodity_Flight_Pairs: Given the *DatasetID* value from Input_OPL_Parameters, OPL pulls a list of commodity-flight pairs from this table. A commodity-flight pair is composed of a *CommodityID* and a *FlightID*,

indicating that the commodity can use that flight (when sharing is allowed). Also included is an indicator that tells whether the commodity's carrier is the same as the flight's carrier. (This data is used when sharing isn't allowed.)

- `Input_Commodity_Airport_Pairs`: Given the ***`DatasetID`*** value from `Input_OPL_Parameters`, OPL pulls a list of commodity-airport pairs from this table. A commodity-airport pair is composed of a ***`CommodityID`*** and an airport, indicating that the commodity is able to visit the airport en route to its destination.

OPL Code

The OPL code is divided into several sections. In the first portion of the model file, tuples are constructed to store the data that is read in from the database. The second part of the code is where the model is defined. Next, there is a main flow control script, where, first, the uninterrupted model is solved and the number of passengers adjusted. Next, the interrupted model is solved. A third LP may be solved if a secondary objective is selected. After the main flow control script, another set of tuples is constructed to export the model's results to the database. For a more in-depth understanding of the code, see the model and the data files.

Output Data Tables

The main output of the model is the passenger flow on each arc. From these values, the total number of shed passengers and the number of shed passengers for each commodity type are obtained. Below is a list of the output tables (in the **`AirlineDataLDRD`** database) and the information they contain:

- `Output_Summary`: For each ***`ModelRunID`***, the ***`DatasetID`*** for the dataset that was used, the values for the five other OPL parameters, the total number of passengers in the system, the total number of passengers after post-uninterrupted solve adjustments, the total number of shed passengers, and the total number of shed passengers that neither originate nor terminate at an interrupted airport are provided.
- `Output_Interrupted_Airports`: For each ***`ModelRunID`***, the set of airports that was interrupted is listed. (Note that the set of interrupted airports is not unique to the ***`DatasetID`***, which is why we include it with the output data. Many model runs may be executed, each using the same dataset, but with different sets of interrupted airports.)
- `Output_Flights`: For each ***`ModelRunID`***, the total number of passengers on each flight during the last LP solve is provided, as well as whether the flight is operational or not (flights in or out of disrupted airports are not operational).
- `Output_Commodities`: For each ***`ModelRunID`*** and commodity, the total number of passengers, the total number of passengers after post-uninterrupted solve adjustments, and the total number of shed passengers from the last LP solve is provided.
- `Output_Commodity_Flight_Pairs`: For each ***`ModelRunID`*** and commodity, the total number of passengers using each flight from the last LP solve is provided.

Historical Data

The main source of historical data is the Bureau of Transportation Statistics website. Three sets of data are obtained from this website: T100 Segment data, OnTime Performance data, and OD Market data. Also, a set of tables was downloaded from the Federal Aviation Administration website and used to map the aircraft tail numbers to the aircraft model and corresponding plane capacity. In addition, a table was constructed that lists the time zone in which each airport is located (for both Daylight Savings Time and Standard Time). All of this data can be found in the `AirlineDataLDRD database on the orca-srn-db\sandbox` server. Below is a list of the six tables of data used for the airline system model:

- `ZZZZ_OD_Market`: Contains the passenger data aggregated at a quarterly level. A given row provides the total number of passengers for each quarter that traveled between an origin and destination, following a specific route, using flights from specific carriers. For example, there would be two separate rows for all passengers flying from Albuquerque to Atlanta to Gainesville, if one set of passengers used Delta Airlines on both flights, and a second set of passengers used Delta for the first leg and US Airways for the second leg. Unfortunately, there is no passenger data at the daily level and no data for passengers flying internationally. This data can be downloaded here: http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=247&DB_Short_Name=Origin and Destination Survey.
- `ZZZZ_OnTime_Performance`: For every flight flown by the major air carriers, this table contains information such as the origin airport, destination airport, and carrier, as well as scheduled and actual arrival and departure times. Also provided is a *`TailNum`* column which (in most cases) can be mapped to a plane model and capacity. This data can be downloaded here: http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time.
- `ZZZZ_T100_Segment`: For each unique tuple (origin airport, destination airport, carrier, aircraft type), this table contains the total number of passengers and seats available. Unfortunately, there is no way of knowing which passengers were flying internationally vs. domestically. This data can be downloaded here: http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=293&DB_Short_Name=Air Carriers.
- `ZZZZ_ACFTREF and ZZZZ_MASTER`: These tables allow most of the values from the `TailNum` column of the `ZZZZ_OnTime_Performance` table to be mapped to an aircraft model and corresponding plane capacity. This data can be downloaded here: http://www.faa.gov/licenses_certificates/aircraft_certification/aircraft_registry/releasable_aircraft_download/
- `ZZZZ_Airport_Timezones`: For each airport appearing in the `ZZZZ_OnTime_Performance` table, the time zone was looked up manually. Therefore, this table may need to be updated if there are new airports appearing in other

years' worth of data that may added to the ZZZZ_OnTime_Performance table in the future.

(The reason the tables all have ZZZZ_ as a prefix is just to keep them listed at the bottom, separate from the tables that are generated for the OPL model input and output, and from the analysis tables).

Model Preparation Queries

Given the six tables above, the *Create_New_OPL_Dataset.sql* query can be executed to manipulate the data and construct a manageable dataset that can then be fed into the OPL model. Before execution, the user must open the query and set the values for 11 different parameters. The user will find the following portion of code near the top of the file:

```
SET @DatasetID=(SELECT MAX(DatasetID) FROM Input_Dataset_Input
_Parameters)+1
SET @DatasetID =   1
SET @StartDate =   '8/10/2011'
SET @StartTime =   400
SET @EndDate =     '8/11/2011'
SET @EndTime =     400
SET @TimePdsPerHour = 1
SET @MinAirportVolumePercentage = 0.2
SET @MinCarrierVolumePercentage = 0.2
SET @MinAirplaneCapacity = 3
SET @MinCommoditySize = 900
SET @NumRoutesToIncludePerODPair = 5
SET @MaxNumLegsForTopRoutes = 3
```

The user must first decide whether to create a totally new dataset or to overwrite a previously generated dataset. If creating a new dataset, the line '`SET @DatasetID = 1`' must be commented out. If overwriting a previous dataset, the user must replace '`1`' (or whatever the line currently says), with the `DatasetID` value for the dataset they wish to overwrite. Next, the user determines the start date, start time, end date, and end time for the time window the user wishes to model (400 means 4:00 am and 2400 means midnight). The user also determines how finely/coarsely the time window is discretized by setting the `TimePdsPerHour` value.

The user controls how large an airport must be (in terms of total inbound and outbound flights) to be included in the model by selecting the value for `MinAirportVolumePercentage`. A value of 0.2, for example, means that only airports that are either the origin or destination of at least 0.2% of the flights listed in the `ZZZZ_OnTime_Performance` table are kept in the model.

Similarly, the user controls how large a carrier must be to be included in the model by setting the value for `MinCarrierVolumePercentage`. A value of 0.2, for example, means only carriers that operate at least 0.2% of the flights listed in the `ZZZZ_OnTime_Performance` table are kept in the model as individual carriers. Those carriers not operating at least 0.2% of

the flights will be combined into an "other" carrier. (For the most part, this value should just remain at 0. For 2011, the on-time performance data only contains the flights for 16 different carriers. Three of these carriers, Mesa Airlines, ExpressJet Airlines, and Atlantic Southeast Airlines, have been chosen to be lumped into the "other" carrier, regardless of their volume percentage. This is done in a different portion of the code.)

By selecting a value for the `MinAirplaneCapacity` parameter, the user restricts the model from including any flights from the `ZZZZ_OnTime_Performance` table that have a plane capacity less than `MinAirplaneCapacity`.

There are many commodities - (origin, destination, carrier) triples – that have very few passengers throughout the quarter. Setting `MinCommoditySize = 900`, for example, restricts commodities from being created that have less than 900 passengers in the given quarter (or roughly 10 passengers/day). To account for these smaller commodities that are not created, the number of passengers in the other commodities is scaled up so that the total number of passengers in the system is not changed.

The last two parameters, `NumRoutesToIncludePerODPair` and `MaxNumLegsForTopRoutes`, are used to restrict the sets of flights and airports that each commodity may use. When `NumRoutesToIncludePerODPair = 5`, for example, for each origin-destination pair, the top five routes (in terms of the total number of passengers flying along that route during the quarter) are selected. Then, in the OPL model, passengers can only use flight legs that appear in those top 5 routes. When `MaxNumLegsForTopRoutes = 3`, the "top routes" must consist of no more than three segments.

Once the user specifies the values for the parameters listed above, the *Create_New_OPL_Dataset.sql* query can be executed. The query is summarized in the following steps:

- Step 0:
  - The values for the parameters mentioned above are stored in a row of the `Input_Dataset_Input_Parameters` table for future reference.
  - The flights from `ZZZZ_OnTime_Performance` that fall within the specified time window are copied into a table named `OnTime_Performance`.
  - The flight data from `ZZZZ_T100_Segment` for the month in which the time window falls is copied into a table named `T100_Segment.`
  - The passenger data from `ZZZZ_OD_Market` for the quarter in which the time window falls is copied into a table named `OD_Market`. Only rows for which the carrier listed in the `TkCarrier` column is one of the carriers from the `OnTime_Performance` table are included (or those with `TkCarrier='99'`, meaning the passengers use multiple carriers).
  - Primary keys are added to these tables.
- Step 1:
  - Rows of the form (`DatasetID`, `Airport`, `AirportID`) are added to the `Input_Airports` table, where `Airport` is the three letter identifier and

AirportID is an integer ID for the airport. Only the airports that are either the origin or destination for at least *MinAirportVolumePercentage* percent of the total flights from OnTime_Performance are added.

- Step 2:
  - Rows of the form (*DatasetID*, *Carrier, Model_Carrier*) are added to the Input_Carriers table, where *Carrier* is the two letter identifier for the carrier and *Model_Carrier* is initialized to be *'OTH'*. A row is added for each carrier that appears in the OnTime_Performance table.
  - The *Model_Carrier* column is updated, with *Model_Carrier* = *Carrier* for any carrier that operates at least *MinCarrierVolumePercentage* of the total flights from the OnTime_Performance table. *Model_Carrier* = *'OTH'* for Mesa Airlines, ExpressJet Airlines, Atlantic Southeast Airlines, and all other carriers.
- Step 3:
  - The data from the T100_Segment table is grouped by carrier, origin, and destination to create the T100_Summary table (over multiple aircraft types). This can be used to estimate plane capacities for the flights from the OnTime_Performance table without a *TailNum* value that can be mapped to an aircraft type and plane capacity.
- Step 4:
  - The Tickets_Split_Out table is constructed by taking the OD_Market data and separating the *AirportGroup* and *OpCarrierGroup* columns each into 9 separate columns ((*Airport1*, *Airport2*, ..., *Airport9*) and (*Carrier1*, *Carrier2*,…,*Carrier9*)). These identify the sequence of airports visited and the carriers used along the passengers' route from origin to destination.
  - Three columns are added: *Model_Origin*, *Model_Dest*, and *Model_Carrier*.
  - For each row, the *Model_Origin* value equals the first airport visited on the passengers' itinerary that appears in the Input_Airports (given the current *DatasetID*).
  - For each row, the *Model_Dest* value equals the last airport visited on the passengers' itinerary that appears in the Input_Airports table (given the current *DatasetID*).
  - For each row, the *Model_Carrier* value equals the value in the *Model_Carrier* column of the Input_Carriers table for the carrier listed in the *TkCarrier* column of the Tickets_Split_Out table (when *TkCarrier = '99'*, then *Model_Carrier = 'OTH'*).
- Step 5:
  - Rows of the from (*DatasetID*, *Model_Origin*, *Model_Dest*, *Model_Carrier*, *Passengers*, *CommodityID*) are added to the Input_Commodities table by taking the tickets from Tickets_Split_Out and grouping all that have the same *Model_Origin*, *Model_Dest*, and *Model_Carrier*. The *CommodityID* column is a unique

integer identifying the commodity (unique only when comparing to other commodities within the **same dataset**).
    - If a commodity has less than *MinCommoditySize* total passengers over the whole quarter, it is not included (after being multiplied by 10 to account for the 10% sampling).
    - Based on the total number of passengers left out because the commodities they compose are too small, the number of passengers for other commodities is scaled up so the total number of passengers remains the same.
    - Also, the number of passengers is scaled by the ratio of the total number of flights from the ZZZZ_OnTime_Performance database that occur during the specified time window to the total number of flights during that quarter (roughly 1/90 if the time window is a single day).
- Step 6:
    - The Trimmed_OnTime_Performance table is created by selecting all flights from OnTime_Performance that originate and terminate at airports from the Input_Airports table (given the current *DatasetID*).
    - A *Model_Carrier* column is added and given a value mapped from the Input_Carriers table (so if the flight is operated by Mesa Airlines, then *Model_Carrier='OTH'*, but if operated by Delta Airlines, then *Model_Carrier = 'DL'*).
    - A *Capacity* column is added and filled in based on a mapping from the *TailNum* column to the *No_Seats* column from the ZZZZ_ACFTREF table. The average capacity from the T100_Summary table is used for flights without *TailNum* values that cannot be mapped to a capacity
    - The *DepTimePeriod* and *ArrTimePeriod* columns are added and populated based on the *CRSDepTime* and *CRSArrTime* columns (the time zone adjustments are also made here).
    - Flights that fall out of the time window after the time zone adjustments are thrown out.
- Step 7:
    - Rows of the form (*DatasetID*, *Model_Origin*, *Model_Dest*, *Model_Carrier*, *DepTimePeriod*, *ArrTimePeriod*) are added to the Input_Flights table by grouping flights from the Trimmed_OnTime_Performance table that have the same *Model_Origin*, *Model_Dest*, *Model_Carrier*, *DepTimePeriod*, and *ArrTimePeriod* (when the number of time periods per hour is 1 or more there are very few instances in which multiple flights are combined; so most flights in Input_Flights actually represent true historical flights instead of combined flights).
    - Only flights with a capacity of at least *MinAirplaneCapacity* are kept.
    - The Trimmed_OnTime_Performance table includes passengers that flew with multiple carriers, some of which may not have their flights listed in the ZZZZ_OnTime_Performance table. Therefore, the total number of passengers flying between each pair of airports on these "unrepresented" carriers is calculated. Based on these results, capacity is added to the existing flights, or if

no existing flight occurs between the airports, a flight is created and added to the list.

- Step 8:
  - The `TopRoutes` table is created, which, for each passenger OD pair, contains the top routes traveled between the airports (ranked by total number of passengers using the route). Only the top *`NumRoutesToIncludePerODPair`* routes that are composed of no more than *`MaxNumLegsForTopRoutes`* segments and are used by at least 1% of passengers over the whole quarter are included.
  - Next, the `Usable_Legs_ByOD` table is created, which lists, for each passenger OD pair, the flight segments (or legs) that may be used to travel from origin to destination. This table is constructed by taking all flight segments from the *`TopRoutes`* table for each OD pair.
- Step 9:
  - The `Usable_Carriers` table is created, which lists, for each *`Model_Carrier`*, the other carriers that its passengers can use. This is based on the historical data. For example, if there are passengers in the `Tickets_Split_Out` table that have Delta Airlines as the *`TkCarrier`*, but use United Airlines as the *`OpCarrier`* for one of the segments of the flight, then we allow Delta passengers to use United Airlines flights in our model (when sharing is allowed).
- Step 10:
  - Rows of the form (*`DatasetID`*, *`CommodityID`*, *`FlightID`*, *`SameCarrier`*) are added to the `Input_Commodity_Flight_Pairs` table based on the `Usable_Carriers` and `Usable_Legs_ByOD` tables.
  - The *`SameCarrier`* indicates if the commodity and flight share the same *`Model_Carrier`*. This is used by OPL in the event that *`UninterruptedSharing`* = 0 or *`InterruptedSharing`* = 0.
  - Any flights that don't appear in a commodity-flight pairing are deleted.
- Step 11:
  - Rows of the form (*`DatasetID`*, *`CommodityID`*, *`Airport`*) are added to the `Input_Commodity_Airport_Pairs` table.
  - This lists all the airports each commodity is allowed to visit, based on the flights it can use from the `Input_Commodity_Flight_Pairs` table.
  - Any airports that don't appear in a commodity-airport pairing are deleted.
- Step 12:
  - Monthly load factors are calculated from `ZZZZ_T100_Segment` for each of the carriers and put in the *`MonthlyT100LoadFactor`* column of the `Input_Carriers` table.
  - Quarterly load factors are calculated from `ZZZZ_T100_Segment` and from `ZZZZ_OD_Market` and put in the *`QuarterlyT100LoadFactor`* and *`QuarterlyODMarketLoadFactor`* columns of the `Input_Carriers` table.
    (The load factors from `ZZZZ_T100_Segment` include passengers that use domestic flights before eventually flying internationally, which lead to larger load

factor values than those that are estimated from the ZZZZ_OD_Market table, which includes only passengers flying domestically.)

- Step 13:
    - All of the temporary tables are deleted

Once the query is completed, a new dataset has been constructed. This dataset can then be used in OPL by specifying its *DatasetID* in the Input_OPL_Parameters table. With a *DatasetID* specified, filtering is done on the Input_Airports, Input_Commodities, Input_Flights, Input_Commodity_Flight_Pairs, and Input_Commodity_Airport_Pairs tables to obtain the desired dataset.

Note: When the OPL code is run, if the number of time periods per hour is greater than 1, then the arrival time period for all flights is increased by 1 time period. This essentially implements a minimum layover time.

For a more in-depth understanding of the query, please see the code.

Post-Processing Queries

After an OPL model is run, in order to do some analysis on the output tables, the *Post_Processing.sql* query should be executed. First, the query must be opened, and a ModelRunID must be specified by editing the following portion of code:

```
SET @ModelRunID = (SELECT MAX(ModelRunID) FROM Output_Summary)
SET @ModelRunID = 1
```

Either select a specific value by using the second line, or comment out the second line to run the query on the most recent model run. Once selecting a value, the query can be executed, and the following tables are created:

- Analysis_Interrupted_Airports: Lists the set of airports that were selected to be interrupted in the current model run being analyzed
- Analysis_Load_Factors: Lists the load factors calculated from the model's solution, as well as the three different historical load factors calculated from the data.
- Analysis_Number_Of_Connections: Provides the total number of adjusted passengers, the number of passengers flying direct, the number of passengers making one connection,*** and the number of undelivered passengers
- Analysis_Undelivered_Passengers: For each OD pair, the total number of adjusted passengers and undelivered passengers is provided.
- Analysis_Undelivered_Passengers_By_Airport: Contains the following seven columns:
    - Airport: The name of the airport
    - AdjustedPassengersOriginating: Total number of adjusted passengers originating at the airport
    - UndeliveredPassengersOriginating: Total number of passengers originating at the airport that are undelivered

- UndeliveredPassengersOriginatingAndNotUsingInterruptedAirport: Total number of passengers originating at the airport that are undelievered and neither originate nor terminate at an interrupted airport
- AdjustedPassengersTerminating: Total number of adjusted passengers terminating at the airport
- UndeliveredPassengersTerminating: Total number of passengers terminating at the airport that are undelivered
- `UndeliveredPassengersTerminatingAndNotUsingInterruptedAirport`: Total number of passengers *terminating* at the airport that are undelievered and neither originate nor terminate at an interrupted airport

*** Given the solution to the multi-commodity flow model, there may be alternative ways of constructing a route-based solution. Because of this, it may be impossible to put an exact count on the number of passengers making a single connection. However, we can calculate the minimum and maximum number of passengers making a single connection.

Validation

Two measures were used to assess the validity of the model. The first was the load factors for the carriers; the second was the distribution for the number of connections made by the passengers.

Load Factors

Given a specific carrier, its system-wide load factor is calculated by taking the total number of passenger-miles (sum over all flights: the number of passengers times the flight distance) and dividing by the total number of seat-miles (sum over all flights: the number of available seats times the flight distance).

Given a solution to the OPL model, the load factors are easily calculated. Typically, the T100 Segment data is used to calculate load factors. This data allows monthly load factors to be calculated for each of the carriers. However, the problem with comparing the model's load factors to the T100 Segment load factors is that the passenger OD table in the model was constructed directly from domestic passenger data. However, the load factors calculated from the T100 Segment data inherently include passengers flying on domestic segments as part of an international itinerary. Therefore, we use the OD Market data (which includes only passengers with domestic origins and destinations) to estimate "domestic" load factors.

The process to estimate the load factors from the OD market data is simple. First, for each carrier, the total number of passengers flying on each segment is estimated by multiplying the passenger sum from the data by 10. Then, the total number of passenger-miles is calculated by summing the passenger number multiplied by the segment distance. The total number of seat-miles is calculated from the T100 Segment data, then the load factor is found by dividing the passenger-miles by the seat-miles.

While these estimated load factors are a better measure for comparison than the T100 Segment load factors, there are still some issues. First, because the OD Market data is only a sample, the exact domestic load factors cannot be calculated. Second, it is unrealistic to

expect the model to provide a solution with historically accurate load factor values for the following reasons:

- The passenger OD table is purely an estimation.
- Some airports may not be included in the model.
- Some passengers (those composing smaller commodities) are not included in the model.
- The capacities for some of the flights are estimated.
- The capacities for some flights are adjusted based on the lack of flight data for the smaller carriers.
- An optimal extreme point solution to the LP results in a large number of empty flights which is unrealistic.

Given these downfalls, we believe it is sufficient to strive for an average load factor deviation (over all carriers) that is no greater than 10-15%. During the development of the model, this goal was usually achieved, especially when setting `MinimizeLoadFactorDeviation` equal to 1.

Distribution for Number of Connections

From the OD Market data, we are able to estimate the distribution for the number of connections made by passengers.

For 2011, this distribution is summarized in the table below:

Table 4: Number of connections made by passengers

# Connections	Percentage
0	65.89%
1	31.51%
2	2.32%
3	0.25%
4	0.02%

Given a solution to the OPL model, there may be multiple ways of constructing a set of routes and a corresponding number of passengers using each route. For example, consider the following network flow model, in which node 1 has a supply of 2 and node 5 has a demand of 2. Assume that there is an optimal solution in which each arc has a flow of 1. When constructing a route-based solution from this arc-based solution, one possibility would be to have 1 unit of flow on each of the routes 1-3-2-5 and 1-4-3-5. Another possibility would be to have 1 unit of flow on each of the routes 1-3-5 and 1-4-3-2-5.

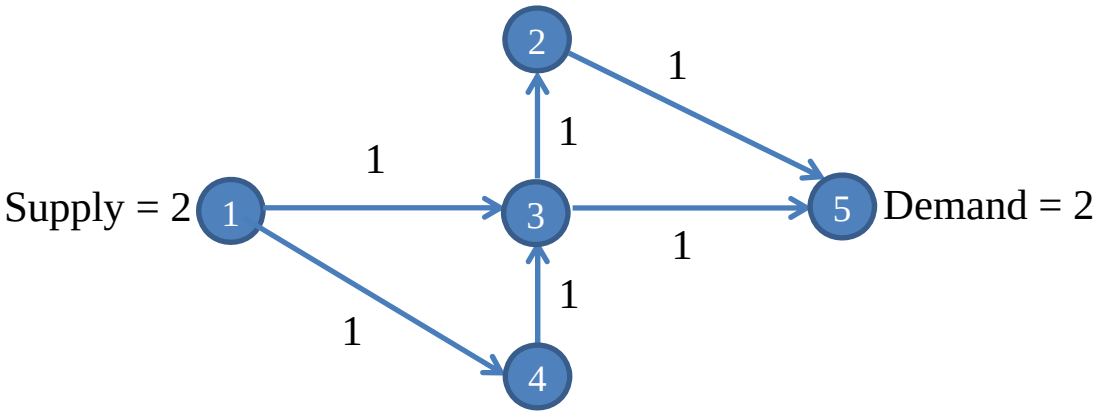


Figure 16. Simple Air Network

Because of this, it may be impossible to calculate the exact number of passengers making one or more connections. However, we are able to calculate the minimum and maximum number of passengers that must make a single connection (in fact, in nearly all studied cases, these values were the same). Because the number of passengers that historically make more than a single connection is so small, and because calculating the minimum and maximum number of passengers making more than a single connection in the model's solution would be quite complicated, we only make comparisons between the number of direct and single-connection passengers. As with the load factor measure, there are several issues with this comparison:

- The historical distribution is merely an estimate, because it comes from the OD Market data, which is a 10% sample.

- When the smaller airports are removed from the model, the number of passengers flying direct increases.
- Without adding some type of penalty term, there is no way to discourage the model from doing strange things, such as having a passenger fly from Albuquerque to Los Angeles to Denver, back to Los Angeles, and then to Atlanta. However, including such a penalty term to restrict such things from happening could create biases that affect the model's validity in other aspects.

Fortunately, while these issues exist, when the majority of airports are included, the model seems to provide solutions very close to the historic distribution for the number of connections made by passengers.

Suggested Parameters Settings

Many different parameter settings were used throughout the development of the model. Ideally, a set of parameter values would exist that provide the most realistic model that is solvable in matter of seconds. Unfortunately, this is not the case. A sacrifice must be made between how realistic the model is and how long it takes to solve. Because it this model is intended to be solved many times using different hazard scenarios, a solution time of no more than a few minutes is desired. The table below shows nine different model runs, each having varying parameter settings. Also shown in the table are the resulting output parameters that describe the resulting dataset, the model size in terms of the number of constraints and variables, the number of shed passengers, the number of empty flights, the largest load factor deviation (over all carriers), and the total computation time (using a Intel Cors2 Quad CPU Q9650 with 3.00 GHz and 8GB of RAM). The time window used for each of the datasets was from 8/10/11 at 4:00 am to 8/11/11 at 4:00 am, and in each model run, 'ATL' was the only airport interrupted.

Note: The total number of airports is not only a function of `MinAirportVolumePercentage`. Even though an airport may be kept after the initial selection, all airports that don't appear in a commodity-flight pairing are removed. Small airports that don't have a large volume of passengers during a given quarter may not be included as the origin or destination of any of the commodities. Also, they may not be used as intermediate airports for any of the other commodities. In such cases, these airports will not be included in the model.

Table 5. Suggested Parameter Values

ModelRunID	1	2	3	4	5	6	7	8	9
Dataset Input Parameters:									
DatasetID	1	2	3	3	3	3	4	5	6
TimePdsPerHour	1	1	1	1	1	1	1	1	2
MinAirportVolumePercentage	0.2	0.1	0	0	0	0	0	0	0
MinCarrierVolumePercentage	0	0	0	0	0	0	0	0	0
MinAirplaneCapacity	3	3	3	3	3	3	3	3	3
MinCommoditySize	4,500	4,500	4,500	4,500	4,500	4,500	4,500	1,800	4,500
NumRoutesToIncludePerOD	5	5	5	5	5	5	10	5	5
MaxNumLegsForTopRoutes	3	3	3	3	3	3	3	3	3
Dataset Output Parameters:									
# Airports	81	117	151	151	151	151	164	217	151
# Carriers	13	13	13	13	13	13	13	13	13
# Flights	13,433	14,842	15,089	15,089	15,089	15,089	15,554	17,102	15,197
# Commodities	4,024	4,319	4,236	4,236	4,236	4,236	4,236	8,593	4,236
# Commodity-Flight Pairs	92,404	107,003	126,232	126,232	126,232	126,232	196,557	264,029	128,394
# Commodity-Airport Pairs	14,277	16,253	18,038	18,038	18,038	18,038	25,510	38,534	18,028
Model Size:									
# Constraints	365,825	415,769	459,155	459,155	459,155	459,155	639,381	950,897	891,803
# Variables	438,365	500,116	560,564	560,564	560,564	560,564	803,210	1,176,139	995,276
OPL Parameters:									
FixUninterruptedCommodities	0	0	0	1	0	0	0	0	0
UninterruptedSharing	1	1	1	1	1	1	1	1	1
InterruptedSharing	1	1	1	1	1	1	1	1	1
MinimizeEmptyFlights	0	0	0	0	1	0	0	0	0
MinimizeLoadFactorDeviation	0	0	0	0	0	1	0	0	0
Solution Output:									
# of Shed Passengers	137,591	126,519	109,650	111,866	109,650	109,650	107,356	109,393	109,560
Empty Flights	2,308	2,898	2,778	3,058	850	3,134	2,559	2,816	2,814
Passengers Flying Direct	78.1%	77.1%	75.6%	76.0%	74.6%	76.0%	67.8%	65.0%	76.0%
Avg Load Factor Deviation	7.7%	9.7%	10.7%	10.8%	10.3%	6.7%	9.6%	9.6%	11.0%
Total Computation Time (s)	70.731	93.077	148.58	310.9	228.51	336.38	542.51	911.063	377.489

Note: The results in this table were obtained before a minimum layover time of one time period was enforced; in addition, a few changes were made that reduce the number of commodities, commodity-flight pairs, and commodity-airport pairs. However, the overall trends remain the same.

Each of the first three model runs uses a different dataset. The only difference between the three datasets is the value for `MinAirportVolumePercentage`. As is expected, when the value for `MinAirportVolumePercentage` decreases, we see an increase in the resulting number of airports that are kept in the model. As a result of a larger number of airports being included, there is a slight increase in the number of flights, commodities, commodity-flight pairs, and commodity-airport pairs. The OPL model takes slightly longer to solve and provides a solution with a better objective function (# of shed passengers), and slightly better percentage of passengers flying direct, but is worse in terms of the number of empty flights and the average load factor deviation.
When comparing model runs 3-6, the same dataset is selected, but different values for the OPL parameters are used. When setting *`FixUninterruptedCommodities = 1`*, the computation time increases and the objective function value is worse, as expected. The increase in computation time is mainly a result of the time that it takes to set the bounds for the variables When setting *`MinimizeEmptyFlights = 1`*, the computation time is significantly greater and the objective function value is the same. As can be seen, there's a great decrease in the number of empty flights, which is clearly more realistic. When setting *`MinimizeLoadFactorDeviation = 1`*, a significant decrease in the average load factor deviation is seen. However, the increased computational burden is probably not worth it.
Model runs 7-9 differ from model run 3 in that the datasets used each vary from dataset 3 by one parameter value. Dataset 4 differs from dataset 3 in its value for *`NumRoutesToIncludePerOD`*. Increasing this value gives the passengers more flexibility with the flights they can use and the airports they can visit. This results in an improved objective function and percentage of passengers flying direct, but the computation time greatly increases. Dataset 5 differs from dataset 3 in its value for *`MinCommoditySize`*. Decreasing this value results a larger number of commodities. Many of these new commodities are probably flying between smaller airports, which results in a decrease in the number of passengers flying direct (because they probably have to fly through at least one large airport en route to destination). While this is desirable, because it gives a more realistic solution, the increase in computation time is very significant and may not be worth it.
Dataset 6 differs from dataset 3 in its value for *`TimePdsPerHour.`* Increasing this value will give a slightly more realistic solution, but is probably not worth the increase in computation time.

Based on these results, the following parameter settings/ranges are suggested:

- `TimePdsPerHour = 1`
- `0 <= MinAirportVolumePercentage <= 0.2`
- `MinCarrierVolumePercentage = 0`
- `MinAirplaneCapacity = 3`
- `3600 <= MinCommoditySize <= 4500`
- `5 <= NumRoutesToIncludePerODPair <= 6`
- `MaxNumLegsForTopRoutes = 3`

Also, it is suggested that the time window be roughly one day in length. It is important to note, however, that as the number of interrupted airports increases, the computation time will

probably increase as well.

Keeping `MinCarrierVolumePercentage = 0` should be sufficient, because there are only 16 carriers with flight data in the On Time Performance data. Three of these carriers, as mentioned before, have been chosen to automatically be grouped in the "other" category, which means 13 individual carriers remain. If for some reason, the user chose to have all carriers act as a whole, then they should set `MinCarrierVolumePercentage = 1`. A few flights appear in the database with a capacity of 1 or 2, and letting `MinAirplaneCapacity=3` restricts these flights from being included in the model. The value for `MaxNumLegsForTopRoutes` is relatively unimportant, because most of the top routes for a given OD pair will use fewer than 4 segments; but,it does provide the user with a little extra flexibility.

The value chosen for `MinCommoditySize` has a large impact on the model size and computational time. Setting `MinCommoditySize = 4500` means that only commodities with at least (roughly) 50 people per day are included in the model (because there are about 90 days each quarter). Decreasing the value of this parameter increases the number of commodities, which greatly impacts the model size and computational time. Letting `MinCommoditySize = 0` is certainly not possible, because it results in hundreds of thousands of commodities. Because the model size is linear with respect to the number of commodities times the number of airports times the number of time periods, this would result in a model with hundreds of millions of variables and constraints.

In addition to the suggestions for the parameters used to create the manageable datasets, the following are suggestions for the values to use for the OPL parameter values:

- *`FixUninterruptedCommodities = 0`*
- *`UninterruptedSharing = 1`*
- *`InterruptedSharing = 1`*
- *`MinimizeEmptyFlights = 0`*
- *`MinimizeLoadFactorDeviation = 0`*

The reason for suggesting to set *`FixUninterruptedCommodities = 0`* is that when *`FixUninterruptedCommodities = 1`*, the lower and upper bounds for a large number of variables must be set during the main flow control script, which often takes a significant amount of time. When *`UninterruptedSharing = 1`* and *`InterruptedSharing = 1`*, code sharing is allowed, which actually does occur between many of the carriers. While the logic we used to allow sharing may lead to more sharing than actually occurs, it is probably more realistic than when *`UninterruptedSharing = 0`* and *`InterruptedSharing = 0.`* When *`MinimizeEmptyFlights = 0`*, a more realistic solution is obtained, but the extra computation time is probably not worth the added realism.

DISTRIBUTION

1	MS1188	Richard O. Griffith	6130
1	MS1188	Dean A. Jones	6131
1	MS1188	Patrick Finley	6131
1	MS9151	Robert L. Hutchinson	8960
1	MS0899	Technical Library	9536 (electronic copy)
1	MS0359	D. Chavez, LDRD Office	1911


Sandia National Laboratories