

Decision Analysis for Toxic by Inhalation Material Routing (DATIMR)



George Mason University
Fairfax, Virginia
OR-680/SYST-798 Dr. Kathryn B. Laskey

09 May 2008

The Railroad Team
Matthew Albin; Ruben Luna; Danielle Martin
Danielle Obuchon; Andrew Ramsay

(Refer to incorporated disclaimer and distribution statements)

Sponsors

Dr. Rajni Goel, Howard University
Mr. Mark Hartong, Federal Railway Administration
Dr. Duminda Wijesekera, George Mason University

The views expressed in this report do not necessarily reflect those of the United States Government, the US Department of Transportation, the US Department of Energy, the Federal Railroad Administration, the Oak Ridge National Laboratory, or their associated contractors/subcontractors. Material in this report shall not be used for advertising or other commercial activities without the express written consent of the United States Government, the Department of Transportation, the Federal Railroad Administration, the Department of Energy, and the Oak Ridge National Laboratory.

This report was prepared for authorized distribution.

TABLE OF CONTENTS

Abstract..... 4

I. INTRODUCTION..... 5

 The Team Members 5

 Sponsors..... 5

 Purpose..... 5

II. PROBLEM DEFINITION 5

 What are the current concerns?..... 5

 Who is Involved? 6

 Existing Route Analysis Tools..... 6

 Where might this Problem Occur?..... 8

 When do such Problems Occur?..... 8

 Why are we Concerned About the Problem?..... 8

 Project Objective:..... 8

III. STRATEGY AND APPROACH..... 8

IV. SOLUTION DEVELOPMENT 15

 Data Collection and Route Selection 15

 The Optimization Model..... 17

 Alternative Approach..... 20

 Consideration of Weights 20

V. RESULTS 21

 Computational Results..... 21

 Stochastic Model Results:..... 26

VI. CONCLUSIONS 29

Appendix A: Operational Requirements..... 32

Appendix B: Project Management..... 35

Appendix C: Deterministic Model..... 39

Appendix D: Stochastic Model..... 50

Appendix E: Weighted Deterministic Model 61

Abstract

In the United States, transportation of toxic by inhalation (TIH) chemicals by rail constitutes a hazard to the general public as can be observed in recent train incidents that resulted in loss of life, injuries, damage to infrastructure, activation of first responder resources, and multiple millions in costs for liability settlements. Furthermore, given the terrorist attack threat, railroads are vulnerable to malicious attacks with potential catastrophic effects to the public, economy, and national security. Carriers (railroads), shippers (chemical industry), federal and local authorities, and the general public have engaged in a national-level initiative to explore options to reduce this security risk. This paper investigates a simplified decision model to select train routes that minimize the security risk of an attack premised on using TIH materials as a weapon. The proposed model identifies basic transport cost, fatality, injury, and remediation as the primary parameters of interest to traffic managers. By combining indications and warning information from intelligence resources with these parameters, the model provides railroad traffic managers with an understanding of the effect of key drivers on route optimality. The model was applied to route analysis for the transport of chlorine between Alexandria, Virginia, and Philadelphia, Pennsylvania. It was found that in high risk situations, a railroad traffic manager should consider re-routing TIH material trains through longer but less populated areas to reduce risk. For low risk situations, the current, shorter route through more density populated areas constitutes a viable choice. This decision support model proposes a probabilistic, deterministic approach for estimating the chances and impacts of incidents and informs a simplified decision tree. An alternative approach suggests a stochastic solution should the deterministic approach prove computationally expensive.

I. INTRODUCTION

The Team Members

The Railroad (RR) Team includes five George Mason University (GMU) graduate students: Matthew Albin, Ruben Luna, Danielle Martin, Danielle Obuchon, and Andrew Ramsay. The Railroad Team was formed in Dr. Kathryn B. Laskey's OR-680/SYST-798 Applications Seminar in February 2008. After review of material presented by Dr. Duminda Wijesekera of past research conducted by GMU in various areas relevant to railroad operations, three sponsors expressed interest in the development of a risk-based re-routing algorithm for railroad transportation of Toxic by Inhalation (TIH) materials. The Railroad Team undertook to study that area and to make contributions to the advancement of solutions to those needs.

Sponsors

The project sponsors include: Dr. Rajni Goel, Howard University; Mr. Mark Hartong, Federal Railroad Administration; and Dr. Duminda Wijesekera, George Mason University.

Purpose

The purpose of this project is to contribute to the sponsor's goal of providing a decision support model for RR community users for selection of alternate rail routes to transport TIH materials based on threat levels, TIH material exposure risks, and costs. A decision support model could simplify evaluation of rail transportation en-route risk by incorporation of cost penalties for re-routing, security risks, and threat factors. It could also provide information on routing options needed to implement visualization techniques for review of complex sets of choices for analysis of alternative routes.

II. PROBLEM DEFINITION

What are the current concerns?

TIH materials are toxic gases and liquids that create severe health hazards to humans if an uncontrolled exposure or release occurs during transportation. TIH material transport accidents in the rail industry have led to the release of high TIH material levels causing deaths, injuries, and property damage. The six most commonly transported TIH materials include: Ammonia, Chlorine, Sulfur Dioxide, Hydrogen Fluoride, fuming Nitric acid, and Sulfuric acid. Approximately 21.6 million ton-miles of TIH materials are transported by rail annually through highly populated areas. For example, approximately 8500 tank cars of chlorine per year are shipped through Washington, DC and pass within two blocks of the US Capitol.

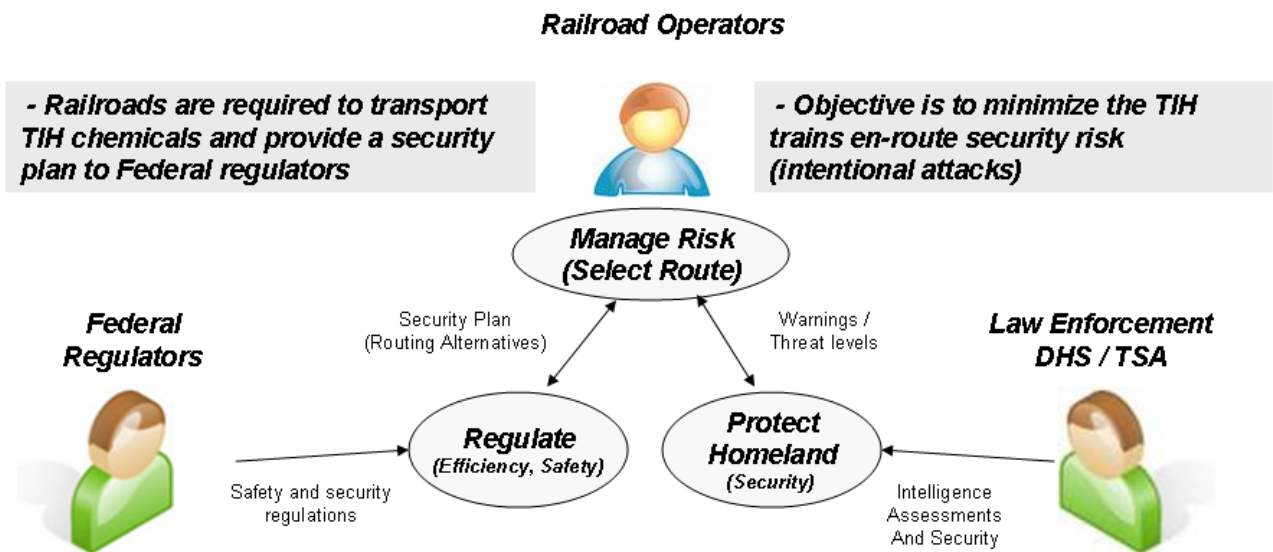
The Federal government requires railroads to transport TIH materials, by means of the railroad's common carrier obligations set by regulations. Railroads typically transport 100,000 carloads of TIH materials each year, virtually all in tank cars. Chlorine accounts for 32,150 tank cars of the 100,000 cars/year¹. While shipments of TIH materials constitute only 3 per cent of all rail carloads and 99.997 per cent of shipments were conducted without incident, when an accident does occur it may have catastrophic consequences.

¹ <http://projects.battelle.org/trbhazmat/Presentations/TRB2007-JA.ppt#336,7>, Freight Rail Snapshot

Today a TIH material release resulting from a terrorist threat is judged highly likely. American Association of Railroads (AAR) standard OT-55-I was approved in December 2006 for transport of chlorine and ammonia by tank cars, which account for 80 per cent of rail TIH movements by the railroad industry. Railroads work closely with the Chemical Transportation Emergency Center (CHEMTREC), a 24 hours/day, 7 days/week operational resource for mitigating hazardous material (HAZMAT) accidents. Railroads have also developed comprehensive Terrorist Risk Analysis and Security Management Plans and are working in coordination with the Department of Homeland Security (DHS) Transportation Security Agency (TSA) to provide data on movement of tank cars containing TIH materials.

Who is Involved?

The three principal end-user stakeholders are operational railroad traffic managers working to plan traffic routes for railroad companies, the Federal Railroad Administration (FRA) in conducting its supervisory responsibilities, and the TSA Railroad Security Task Force (RRSTF) in conducting security analyses and advisory functions. The National Transportation Safety Board (NTSB) could be an occasional user based on its investigatory responsibilities to conduct forensic examinations of hazardous railroad events. Stakeholder relationships are shown in Figure 1.



Existing Route Analysis Tools

Prototype route-planning tools have been created to automate route traffic planning, however, they do not provide operationally useful interpretations of complex routing solutions, and they do not address cost factors pertaining to routes for the decision-maker. Currently routes are selected on a case-by-case basis depending on schedule and shipment types: general freight, scheduled runs, and passengers. There are two existing route analysis tools: the Transportation Routing Analysis Geographic Information System (TRAGIS), and the Rail Routing and Visualization

Application (RRVA), which accesses the TRAGIS routing engine. RRVA provides a basic framework for generation and display of alternative routing information and provides an initial set of capabilities. By gaining knowledge about how these tools operate, we can provide decision-makers with recommendations for optimal route selection based on specific preferences.

TRAGIS – Transportation Routing Analysis Geographic Information System

TRAGIS is a client/server, geographic information system (GIS)-based transportation routing and analysis computer model. The routing engine resides on a central server and receives routing parameters from a specialized client across the Internet or other network path. The output is sent over the network to the client user interface for display. TRAGIS also uses GIS functions to map calculated routes and allows for selection of the route origin and destination from a list of node names. TRAGIS also provides functions to temporarily modify the route network, such as adding states to avoid. Population data for transportation routes is also provided by TRAGIS (see Figure 2).

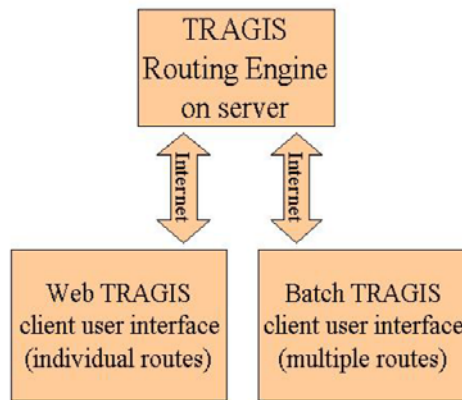


Figure 2. TRAGIS Client/Server Structure

RRVA – Railroad Routing Visualization and Analysis Tool

RRVA utilizes the TRAGIS routing engine. It is designed to assist the FRA Office of Safety in analyzing rail routes within the continental United States. RRVA is a client-server application where the user interface and map data files reside on the user’s client and the routing engine and network data files reside on a network server (see Figure 3).

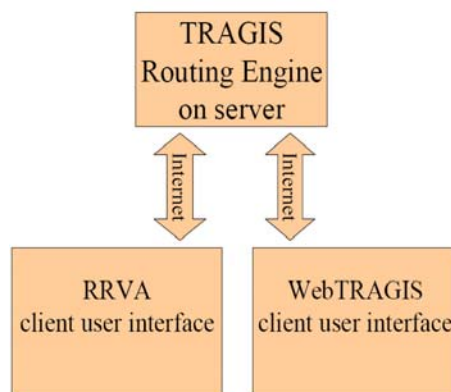


Figure 3. RRVA Top-Level Structure

Where might this Problem Occur?

TIH material transport accidents can occur anywhere. Some of the most severe U.S. TIH material releases in rail transport cases have occurred in South Carolina, Texas, and Montana. Attacks premised on use of TIH materials as weapons are most likely to occur in areas with high population densities such as Washington D.C., Baltimore, Philadelphia and New York City. TIH material release exposure levels will affect a larger quantity of people in highly populated urban areas.

When do such Problems Occur?

As in TIH material rail accidents, attacks can occur at any time and usually without notice. In terms of this project and model, the Railroad Team has made the assumption that if the threat levels determined by the TSA or DHS are high or severe, then there is a higher chance of a terrorist attack, which could mean that a TIH material release is much more likely to occur.

Why are we Concerned About the Problem?

TIH material release and exposure is a threat to human life. By providing train re-routing alternatives, we can significantly reduce the risk of human injury and death if an attack occurred and TIH materials were released. One estimate is that complete release of one 90-ton TIH rail car in the center of Washington, DC could cause death and injury to more than 100,000 people.

Project Objective:

Provide capabilities that enable decision-makers to make informed choices concerning selection of alternate routes for transport of TIH substances based on minimizing risks.

III. STRATEGY AND APPROACH

There has been increased concern over intentional releases of TIH substances. Transport of hazardous material shipments can provide opportunities for terrorists to plan an attack where a release of TIH material could injure or kill people or disrupt the economy². A TIH material release in the context of a terrorist act is effectively a weapon of mass destruction (WMD). A particular concern is that TIH material shipments will be targeted or seized and current safety transportation procedures may not sufficiently manage the risk posed by intentional release of hazardous materials. Therefore, the RR community has developed a research and development program to explore potential solutions to the problem. Figure 4 depicts the technical approach steps we have taken in this project. Although depicted sequentially, its implementation required feedback loops to update the project knowledge base to incorporate newly acquired knowledge, team agreements, and time management strategies.

Step 1- Elicit requirements and research – This step included coordination with project sponsors to define the project scope, problem statement, and expected deliverables. In addition, the team performed research to review available information sources and relevant literature.

² National Transportation Safety Board, Collision of Norfolk Southern Freight Train 192 With Standing Norfolk Southern Local Train P22 With Subsequent Hazardous Materials Release at Graniteville, South Carolina January 6, 2005, Railroad Accident Report NTSB/RAR-05/04, November 2005

Step 2- Develop route selection model – This step synthesizes source information and acquired knowledge to develop an initial description of the decision problem. The effort consisted of defining the problem context and constraints, the potential decision options, set of relevant data, and risk calculation model.

Step 3- Identify attributes and collect data – This step focused on identifying data sources, deciding on an adequate representation of the problem under study by selecting the minimum input data set (relevant factors), and seeking validation from stakeholders and project sponsors.

Step 4 - Select optimization technique – This step implemented operations research techniques to explore a cost function and an analytic solution to the problem. Cost function output is a crucial input to the decision model payoff data equipments.

Step 5- Perform sensitivity analysis – This step consists of performing sensitivity analysis to obtain guidance needed to model a decision problem. The expectation is to use sensitivity analysis output as guidance for successive iterations through the decision analysis cycle (modeling steps, followed by sensitivity analysis, followed by more modeling, and so on). This step is the first iteration in a dynamic cycle of modeling and analysis that strives to arrive eventually at the requisite decision model and to analyze it to understand clearly which alternative should be chosen.

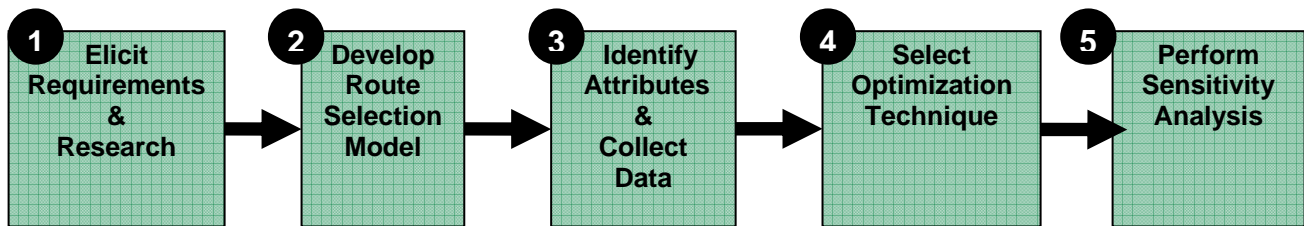


Figure 4. DATIMR Technical Approach

1- Step 1 - Elicit Requirements and Research

Many stakeholders are involved in the TIH material transport problem: carriers (railroad companies), shippers (mostly chemical companies), federal authorities (law enforcement and regulators), and the general public³. The major Federal regulatory authorities include: the Department of Transportation (DOT) Pipeline and Hazardous Materials Safety Administration (PHMSA), FRA, and TSA. All three regulatory authorities are adding new requirements to address the security risk of transporting TIH materials. The DHS is also considered a regulatory authority in terms of their responsibility to develop risk management policies to assess risks in various sectors. Most TIH material transport related information is treated as sensitive security information by the railroad companies and is not available to the public. This information security practice created barriers in terms of the team's ability to incorporate more realistic data and prompted simplification of model development.

³ AAR : Recommended Railroad Operating Practices for Transportation of Hazardous Materials OT-55-I

The TIH material rail-transport problem can be seen as part of a larger enterprise-level risk management framework that addresses financial, operational, engineering, safety, and security considerations⁴. Risk is the quantification of undesirable events in a specific time interval, and in this context relates to the scenario of intentional attacks by terrorist actors on trains transporting TIH materials seeking to inflict maximum damage (Hartong et al 2007)⁵. At this point in history, such events have not materialized yet, but represent a scenario that is of great concern to organizations with responsibilities for risk management⁶. These events represent point values that rarely occur and are also extreme events when compared to day-to-day operational mishaps such as accidents. Current risk assessment methods focusing on intentional attacks or terrorist threats to the rail system are based on collecting subjective probabilities from intelligence community subject matter experts (SMEs) based on their opinions and collected intelligence. These results are disseminated in the form of indications and warnings to the rail network operational staff for near-term implementation of security countermeasures.

In the United States, there are two risk management frameworks that address TIH material train transport security risk problems. The DHS TSA, law enforcement organizations, and railroad industry security organizations attempt to protect and defend the public and rail infrastructure assets, and to enforce laws related to terrorist threats. The DOT PHMSA and the FRA promote a safety framework that minimizes the potential impact of intentional attacks on TIH material train transportation. The DHS and the DOT are concerned about risks posed by the rail transportation of bulk TIH materials in High Threat Urban Areas (HTUA).

The DHS TSA approach to security assesses the risk and evaluates the vulnerabilities associated with different components of the rail systems to balance risk with resources. TSA initiatives are intended to focus the collective limited resources available on prevention of terrorist attacks with the greatest potential consequences. The TSA concept requires that all railroad freight carriers that operate in HTUA will develop annexes to their security plans that are site-specific to that HTUA as defined by DHS Urban Area Security Initiative (UASI) geographic areas. These security plans are developed to be risk-based and include metrics that reflect population density, the amount of TIH materials transported by rail, and the length of time that these shipments are in HTUAs.

The TSA defines risk as a function of population density, number of TIH material shipments, and the length of time TIH material tank cars are unattended and unsecured. Risk is also assessed in terms of the time that TIH material tank cars are held in yards, terminals, on railroad-controlled leased tracks and the time that TIH transport trains are stopped or standing within a HTUA. The TSA rail security goal and strategy is to achieve a 50 per cent reduction in the objectively measured risk of TIH material rail transport by the end of 2008 by creating secure chains of custody of shipments; and minimizing standing, unattended TIH material tank car and

⁴ S.K. Peterson (et al). A Framework for Modeling Rail Transport Vulnerability SFTA Research Report # 24, August 2007.

⁵ M. Hartong, R. Goel, D. Wijesekera. A Risk Assessment Framework for TIH Train Routing, 2007

⁶ H.H. Willis et al, Terrorism risk modeling for intelligence analysis and infrastructure protection, RAND Center for Terrorism Risk Management Policy for the Department of Homeland Security, 2007

trains in HTUA. These plans will be classified appropriately to protect sensitive information. The TSA will provide the rail industry with a list of identified HTUAs.

PHMSA and FRA promote a safety framework that minimizes the potential impact of intentional attacks on TIH material transport and recently proposed changes to DOT/PHMSA 49 Code of Federal Regulations (CFR) Sections 172 and 174, to include guidance that rail carriers are required to analyze the specific safety and security risks for routes. CFR requirements include:

- Route analyses will be required to be in writing and must consider, at a minimum, a number of factors specific to each individual route; and must ensure each route issued for the transport of the specified hazardous materials presents the fewest overall safety and security risks.
- PHMSA proposes systematic processes for rail carriers to (1) Identify the routes currently in use by the rail carrier; (2) Perform safety and security risk analyses of those primary routes; (3) Identify and analyze commercially practicable alternative routes; and (4) Make future route selections based on the results of the completed analyses.

PHMSA expectations are that if an existing route presents the lowest overall safety and security risk, it continues to be the preferred route. The concept establishes that several alternative route selections will be based on the lowest overall safety and security risks. The alternate route will be used depending upon conditions associated with transport requirements, and conditions warranting the route change must be clearly identified in the analyses and routing decision documentation. Also, based on the route analyses, either the existing or alternative practicable route is identified as presenting the lowest overall safety and security risks; however, the rail carrier may identify measures to mitigate risks and lower the overall risk of the other route. Reports must identify the route with the lowest overall safety and security risk and must make recommendations as to what route should be selected and used. Finally, rail industry operators are asked to coordinate and comply with TSA-led Rail Corridor Assessments and the FRA's security plan compliance inspections to include system security, access control, and en-route rail security action items.

Operational requirements for TIH material transport are based on development of a decision support model in the context of risk-based planning, a simplification of the current rail transportation model, and the incorporation of security risk factors.

The Railroad Team acquired data and research through detailed literature review, research and communication with the stakeholders and RR community. The RRVA tool was also used to characterize TIH material transport routes⁷. The project results include the optimization model and software program, decision trees, and two sensitivity analyses for model comparisons.

⁷ P. E. Johnson et al, Railroad Routing and Visualization Analysis (RRVA) User's Manual, Oak Ridge National Laboratory for the Federal Railroad Administration, 2006

2- Develop Route Selection Model (decision model)

Decision analysis consists of a framework and a tool kit for dealing with difficult decisions. Structuring a decision problem may be useful for understanding its precise nature, for generating alternative courses of actions, and for identifying important objectives. The objective of decision analysis is to help the decision-maker. The decision analysis cycle best describes the overall process, which may go several iterations before a satisfactory solution is found (see Figure 5). Work conducted in the DATMIR project constitutes a first decision cycle iteration.

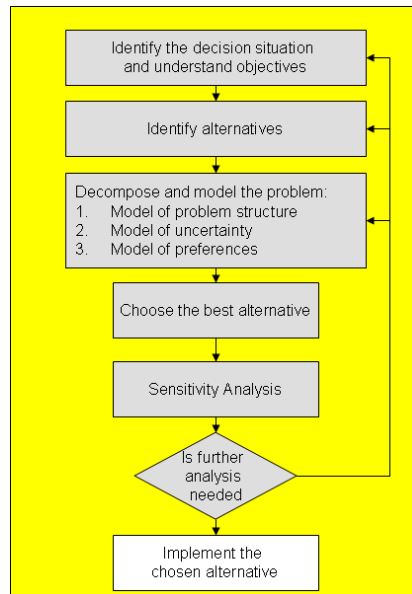


Figure 5. Diagram for a generic risk analysis process

The scenario of intentional attacks on trains to intentionally release hazardous materials has no precedent in U. S. history and for this reason this hypothetical problem becomes a point value event that rarely occurs. This is an extreme event as compared to typical day to day operational mishaps⁸. For this reason, statistical approaches fail to adequately represent catastrophic events given their low probability of occurrence (very sparse) and associated extreme impacts (well outside rail system design goals)⁹. When considering TIH material transport en-route risk, a common objective-function to estimate risk consists of the product of the frequency (probabilities) and consequences of incidents along an entire train route¹⁰.

$$\text{Risk} = (\text{Probabilities}) \times (\text{Consequences})$$

⁸ Christopher P.L. Barkan. Risk Considerations of Railroad Routing of Hazardous Materials: The Appeal & the Dilemma, Railroad Engineering Program University of Illinois at Urbana-Champaign TRB Panel on Rail Routing of Hazardous Materials January 2006.

⁹ Christopher P.L. Barkan. Risk Considerations of Railroad Routing of Hazardous Materials: The Appeal & the Dilemma, Railroad Engineering Program University of Illinois at Urbana-Champaign TRB Panel on Rail Routing of Hazardous Materials January 2006.

¹⁰ M.R. Saat, C. Barkan. The Effect of Rerouting and Tank Car Safety Design on the Risk of Rail Transport of Hazardous Materials, Railroad Engineering Program, University of Illinois at Urbana-Champaign, Urbana, USA

In the process of developing a train routing decision model a risk function is incorporated in a decision model by following three fundamental steps:

- Identify and structure the values and objectives.
- Structure the elements of the decision situation into a logical framework.
- Refine and make the definition of all elements of the decision model more precise.

The DATMIR project developed a fundamental objectives hierarchy in order to identify and validate with project sponsors the most important values to the decision-maker. This model specifies ways to measure the accomplishment of the decision-maker’s selected objectives by creating a visual representation and structure of the objectives to help the team determine the most important factors and variable domains for the risk algorithm. The result of this process was the selection of a minimum set of objectives to support the development of a metric framework for future decision analysis work. The minimum set of objectives include the incorporation of TSA threat indications and warnings, loss of life and injuries associated to a potential attack, and the transportation cost penalty associated with re-routing train through longer distance paths in order to avoid high population exposure (see Figure 6).

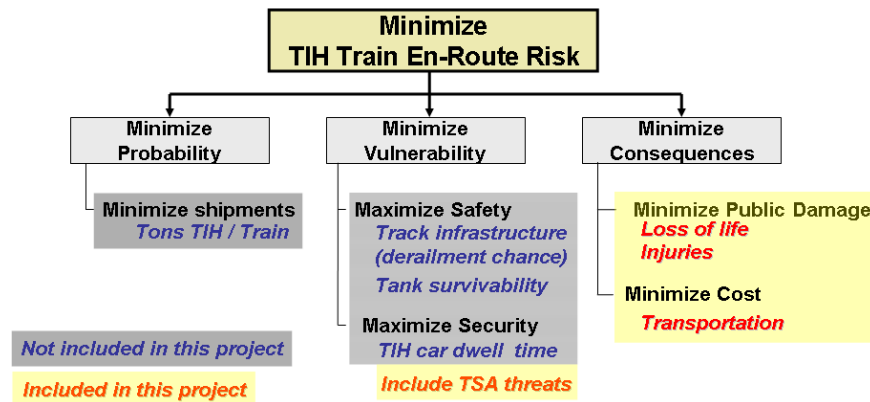


Figure 6. Fundamental Objectives Hierarchy

Hartong et al 2007¹¹ presents two main approaches to measuring TIH material transport risks. The total route risk method measures risk as based on identification of the worst case scenarios along each potential route. The other method is the route segment risk calculation that requires more data to measure risk by route segment and then summing all segments for a given route. In this case probabilities and consequences are estimated at the segment-level.

Before addressing risk functions, a decision model to serve as a generic framework for the decision analysis process is designed. For this the DATMIR project explored applicability of decision tree concepts as a tool to fuse all information and present it in a compact and comprehensive way. A decision tree represents all of the possible paths that the decision-maker

¹¹ Op. cit.

might follow through time, including all possible decision alternatives and outcomes from chance events.

Figure 7 graphically introduces some of the concepts, diagram structure, assumptions, and limitations of the first decision tree developed. The first decision tree was designed for simplicity as an initial baseline for future development. The decision tree name is *TIH Material Routing Decision Tree* and consists of the selection of two main options. The first decision is that the railroad operator accepts the TSA train routing concept or rejects it by always selecting the most practical train route that is commonly the one with the greatest exposure to loss of life. The TSA train routing concept suggests the option of re-rerouting TIH material transport trains through longer distance paths but less populated areas as the threat indication and warnings of potential attacks increase. The second decision is embedded in the chance node description by allocating the estimated risk for all available routes in ascending order as the threat level increases.

The proposed decision tree calculates the Expected Monetary Value (EMV) of the chance node by multiplying chances by the estimated risk for each chance branch and then summing them for each decision option (in this case for one chance node). The EMV can be used to investigate the percentage reduction risk for one year time interval in comparison to the do-nothing decision option (always use the most economical train route in terms of haul cost).

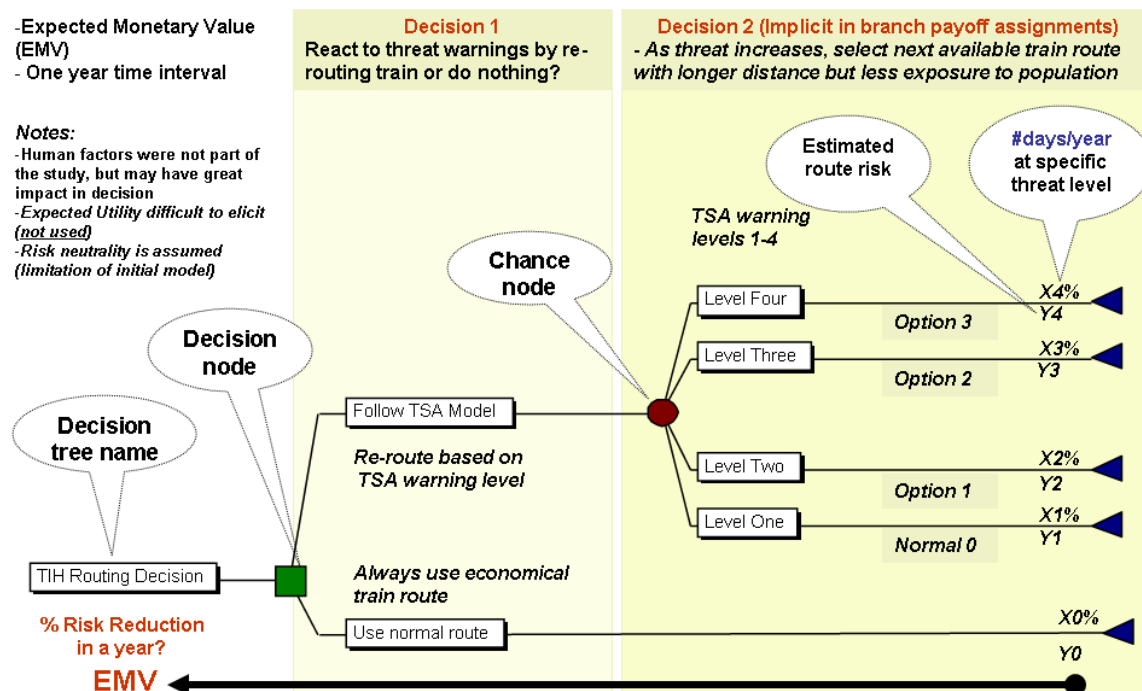


Figure 7. TIH Material Routing Decision Tree

This initial decision tree does not include social factors and moral considerations that may influence decision-maker selection. For instance, re-routing a train equates to an arbitrary risk transfer to rural areas and away from main cities, and suggests inequalities in the value of life as

a function of geographic location (rural versus city). Another limitation of this initial decision tree is the lack of a risk function to capture the decision-maker’s attitude towards risk. This is due to the complexity of the problem and reluctance of stakeholders to disclose their preferences and business practices. Project sponsors advised the team to simplify the problem by using an EMV that indirectly assumes risk neutrality as opposed to the expected utility method that incorporates risk functions.

The outcome of decision analysis is to provide courses of actions for the decision-maker to accept or deny. The decision/risk analyst’s responsibility is to develop a decision framework that is validated by the customer and that provides the best available information for making an informed decision. This is an iterative process that requires constant feedback from customers and all stakeholders involved in the risk management process.

IV. SOLUTION DEVELOPMENT

Data Collection and Route Selection

After developing our algorithm and determining which variables would be included, it was necessary to identify a test scenario and collect input data. The project sponsors felt it was important to select a simple scenario where a few routes could be compared over the course of the semester. The sponsor advised the team to investigate the routes between Philadelphia, PA and Washington, DC. Using RRVA the team was able to identify the possible routes between this origin and destination. These routes are shown in Figure 8.

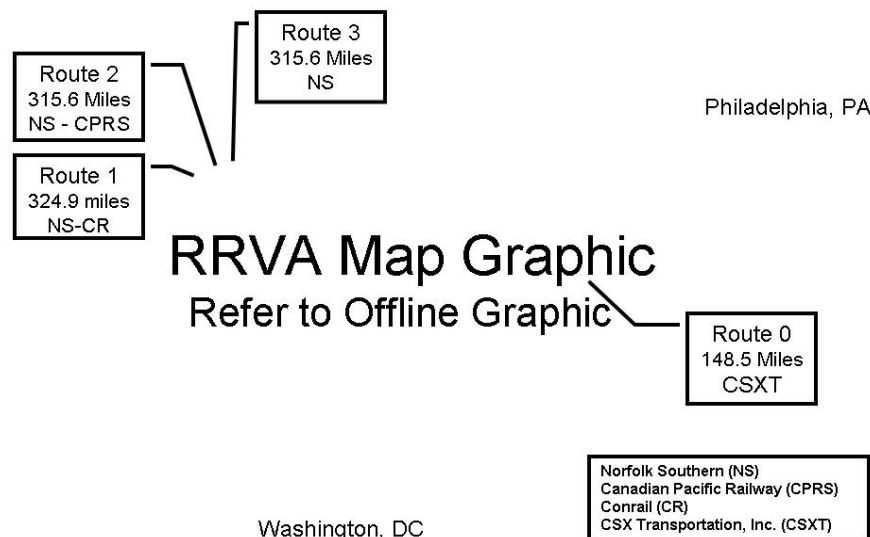


Figure 8. RRVA Routes between Philadelphia, PA and Washington, DC

There are four routes, three of which are very similar and appear superimposed in the graphic above. Routes 1, 2, and 3, shown in purple are significantly longer than Route 0. Route 0 is the

shortest path between the origin and destination, passing through Baltimore, MD. The other three routes avoid Baltimore by traveling on a northwest path.

RRVA provided additional data associated with each of the possible routes between Philadelphia and Alexandria, VA. Each route is composed of segments. In our case, each of the four routes under examination is made up of 50-75 segments. RRVA provides the name of each segment expressed as the city or county in which it lies, or the railroad node name. Each segment varies in length. The distance each segment covers is reported in miles. RRVA also reports the carrier operating along each segment of a particular route.

RRVA provides a relative population density per segment by state; however we required more specific numbers. To best capture the segment population, we used the 2000 US Census report¹². Each segment name was mapped to its respective county or city in the Census Report. The average population density per square mile listed in the census data was used for each the corresponding segments.

Economic factors are a value-added component of our work as they are not accounted for in existing tools. A great deal of research was required to obtain cost data to inform our algorithm. The cost of shipping chlorine was obtained from the RR companies. Many companies have cost calculation tools on their website; however given that chlorine is a high risk substance, this commodity is rarely listed for public access. When possible haul cost or the cost to ship one car of chlorine one mile was obtained from a customer service representative in the chemical department of each RR Company or from pricing tariffs posted at their websites. Chlorine haul costs for Norfolk Southern¹³ and CSX Transportation¹⁴ were obtained in that manner. With the approval of the project sponsors, our team had to make some assumptions about the other two carriers operating along the identified routes. Conrail is operated by CSX and Norfolk Southern. In our routes, Conrail will assume the rate of the joint carrier for that particular route. Canadian Pacific Railway was unable to release any pricing information. In this case we assumed the haul cost to be an average of the haul costs identified for Norfolk Southern and Union Pacific. The price to haul one car of chlorine one mile ranges from \$18-\$25 for the carriers on the routes we are modeling. Table 1 depicts the haul cost data.

Rail Company		Commodity	Shipping Cost per Car/Mile
Canadian Pacific	CPRS	Chlorine	\$21.37
Conrail (handled by CSX and Norfolk Southern)	CR	Chlorine	assume rate of joint carrier on route
CSX Transportation, Inc.	CSXT	Chlorine	\$17.98
Norfolk Southern Corp.	NS	Chlorine	\$24.77

Table 1. Railroad carrier haul cost

It was especially challenging to find data on injury and environmental cleanup costs of chlorine release incidents. After much research, the worst U.S. chlorine accident was used as a guide in

¹² U.S. Census Bureau, Current Population Survey, August 2000.

¹³ Norfolk Southern Public Price Publications <http://www.nscorp.com/mktgpublic/publicprices> CHLORINE GAS 2812815 Public Price Publications for Authority: NSRQ65579

¹⁴ www.csx.com/com/priceFinder.cfc?method=buildDOC&ri=004235543

determining our data functions. On January 6, 2005, a Norfolk Southern train crashed into a parked train in Graniteville, South Carolina as the result of a switching error¹⁵. A tank car ruptured in the accident releasing about sixty tons of chlorine. Nine people died from inhalation of chlorine vapors and hundreds were injured¹⁶. In the settlement agreement, injury compensation ranged from ten to thirty thousand plus special cases which were settled individually¹⁷. Assuming that injury severity is a function of the concentration of chlorine, we used the following function:

$$\text{Cost}_{\text{injury}} = \log (\text{volume}_{\text{released}}) * 18000$$

In the Graniteville, SC accident 5400 residents within a mile of the crash site were evacuated for two weeks while the area underwent decontamination. The estimated site clean-up cost was between three and four million and secondary remediation costs exceeded six million dollars. Using this data, our team estimated the cleanup cost to be about forty five thousand dollars per ton of chlorine released.

The cost of human life will always be a controversial subject. In our research we found that the cost of lives varies greatly from incident to incident. In our base case, we used \$3.7 million per person which is the US Government calculation for the value of a human life of people under the age of seventy¹⁸.

Railroad companies can use a risk-based TIH material transport rail routing algorithm to provide insight to the important objectives and trade-offs of minimizing the consequences of a TIH material release due to terrorist acts. Correspondingly, federal regulators provide guidance on requirements and expected risk management program performance. The intelligence community provides information on potential terrorist threats. Two approaches have been defined to assist the railroad traffic manager in making a more informed decision when selecting an optimal route.

The Optimization Model

This study's goal was to develop a method for displaying to a railroad traffic manager the effects of the most significant parameters on route optimality. The four parameters selected as the most significant, namely haul cost, fatality, number injured, and remediation cost, determine the overall risk cost of a route. Additionally, the optimization model was designed in such a way as to facilitate sensitivity analysis. Two approaches were developed and a third considered but found infeasible.

The most statistically sound approach was the following deterministic program.

¹⁵ http://www.fra.dot.gov/downloads/safety/Accident_Investigation/2005/hq_2005_02.pdf

¹⁶ http://chronicle.augusta.com/stories/052505/met_4237980.shtml

¹⁷ <http://nsrcsettlement.com/documents/Proposed%20Settlement%20Agreement.PDF>

¹⁸ U.S. Environmental Protection Agency, 2003

$$\text{Risk Cost}_{\text{route}} = \sum_{\text{all segments}} [\text{cost}_{\text{car-mile}} * \# \text{cars} * \text{distance} + P(\text{attack}) * P(\text{attack occurs on segment} | \text{attack}) * [\# \text{killed} * \text{cost}_{\text{lives}}(\# \text{killed}) + \# \text{injured} * \text{cost}_{\text{injury}}(\text{volume}_{\text{released}}) + \text{volume}_{\text{released}} * \text{cost}_{\text{remediation}}(\text{volume}_{\text{released}})]]$$

The first product in the risk cost equation, $\text{cost}_{\text{car-mile}} * \# \text{cars} * \text{distance}$, calculates the basic, physical cost of transport by multiplying the cost of transportation of a tank car of chlorine one mile ($\text{cost}_{\text{car-mile}}$) by the number of tank cars of chlorine and the distance of the route in miles.

The probability of attack, $P(\text{attack})$ is the average probability of attack estimated on a per-shipment basis given a particular TSA Railroad Security Task Force Threat Level. The TSA Railroad Security Task Force has four threat levels, which indicate railroad sector-specific attack probabilities. Level 4, the highest level, corresponds to the non-sector specific Homeland Security Level Red and indicates a confirmed threat of attack against the railroad industry or an actual attack in the United States. It is implemented for 72 hours at a time at which point the situation is reevaluated. Railroad Security Task Force Threat Level 3 indicates a credible threat to the US or railroad industry infrastructure. It maps to Homeland Security Level Orange and is renewed every one to two months. Level 2 approximates a 30 per cent chance of an industry attack and is established for several months to years. Level 1, approximating a 20 per cent confidence of attack and is established for several months to years, has never been used. Both Level 2 and Level 1 correspond to Homeland Security Level Yellow.

The probability that an attack occurs on a given segment given that an attack has occurred, $P(\text{attack occurs on a segment} | \text{attack})$, is determined by the proportion of segment population density to sum of all segment population densities. Using the population density of a segment allows segments of different lengths to be evaluated objectively. It also prioritizes segments according to a terrorist’s objective of incurring max human exposure. At the same time, it accounts for the attractiveness provided by targets in less populated that have correspondingly less security. Using the proportion of a segment’s population density to the sum of the population densities ensures that the sum of the probabilities that each segment is attacked (given that there is an attack) sums to one.

The fatality, or number killed, is the product of the probability of death given that there is an attack ($P(D/A)$) and the population exposed ($\text{Pop}_{\text{exposed}}$):

$$\# \text{Killed} = P(D/A) * \text{Pop}_{\text{exposed}}$$

The probability of death is further broken down into a model of a plume over the release site of the attack:

$$P(D/A) = [1 / (1 + e^{-((L_{\text{does}} - \mu) / \sigma)})]$$

where L_{does} is the exposure, in parts per million (ppm), to which people within a hemisphere of an 800 meter radius around the point of release are exposed. The mean (μ) is the mean lethal dosage for a population with a typical age distribution, 210 ppm. Since research did not yield a

standard deviation for human fatality, σ in the above equation, a Gaussian distribution was assumed and the standard deviation for human fatality σ was determined as follows.

$$\int_0^{210} (1/(\sigma\sqrt{2\pi})e^{-(x-\mu)^2/2\sigma^2})dx = 0.5$$

I. Assuming $x-\mu = 0$ yielded $\sigma = 420/\sqrt{2\pi} \sim 167.56$ ppm.

As previously stated, this plume model supposes a hemisphere with an 800 meter radius around the point of release, which is located at a point along the track. Within this hemisphere, the exposure, and therefore the fatality, decreases exponentially the further an individual is from the site of release. Furthermore, the population exposed within the hemisphere was assumed to be distributed uniformly. Two areas for future research would be the fact that this model is limited to a hemisphere with an 800 meter radius. In reality, the size of the area affected is a function of the volume released and other factors. Second, the release cloud typically is not best described by a hemisphere; rather, non-vertical release angles and wind give different shapes to the plume.

The scope of the project, however, precluded further research on plume behavior. Instead, positing an even distribution of airborne chlorine particles within the hemisphere of an 800 meter radius, the Ldoes, or the dosage of chlorine to which an individual is exposed, is the amount released per volume in ppm. The amount released is the product of the number of tank cars of chlorine; the probability that a car will rupture given that there is an accident; the amount of chlorine in an average tank car, namely 90 short tons; and the percentage released given that a tank car ruptures. The probability that a car will rupture given that there is an accident and the percentage released given that a tank car ruptures were engineering problems considered beyond the scope of this paper; therefore, the former was estimated at 33% and the latter at 50%. The volume released in an attack is thus:

$$\text{volume}_{\text{released}} = (\text{number of tank cars}) * (\% \text{ of cars that rupture}) * (\% \text{ contents released}) * (90 \text{ tons/tank car}).$$

The following calculates converts exposure in parts per million within the proscribed hemisphere.

$$\text{exposure}_{\text{ppm}} = \text{volume}_{\text{released}} * (907184.74 \text{ g/ton}) * (1 \text{ L Cl}_2/\text{gCl}_2) * (1000000 \text{ }\mu\text{L Cl}_2/\text{L Cl}_2) / [(2\pi/3) * (800 \text{ meter radius})^3 * (1000 \text{ L}/\text{m}^3)]$$

A route segment is assumed to have a uniform population distribution within an 800-meter band along the track. The population exposed is calculated simply by multiplying the 1600 meter by 1600 meter area by the population density per square meter. Those not killed according to the probability of death given an attack ($P(D/A)$) are assumed to be injured.

The cost of death, the cost of injury, and the cost of remediation are all assumed to be dependent on the number of fatalities or the amount released. Unfortunately, historical data was not readily available to determine the relationship between the numbers of deaths and cost per life, nor between the amount released and the cost of remediation. Therefore, a linear function for these two costs was assumed. Historical data showed, however, that the cost per injured individual varied logarithmically with the amount of chlorine released.

This optimization model allows the calculation of the risk costs for each route under consideration to determine the optimal route with the minimum risk. Furthermore, sensitivity analysis performed on the number of cars allows further insight into a route’s optimality. Additionally, a variable can be substituted into the equations for different routes to determine its effect on route optimality. For example, for a given threat level and number of cars, the cost of life can be treated as a variable to determine for what cost interval one route maintains its dominance over another.

Alternative Approach

Before the deterministic approach described above was fully developed, a stochastic approach was used to determine the risk cost for each route. The following approach introduces stochastic variance into the risk cost computation, but it may be more scalable for analyzing routes with a computationally prohibitive number of route segments.

$$\text{Risk Cost}_{\text{route}} = \sum^{\text{all segments}} [\text{cost}_{\text{car-mile}} * \# \text{cars} * \text{distance}] + P(\text{attack}) * [\# \text{killed} * \text{cost}_{\text{lives}}(\# \text{killed}) + \# \text{injured} * \text{cost}_{\text{injury}}(\text{volume}_{\text{released}}) + \text{volume}_{\text{released}} * \text{cost}_{\text{remediation}}(\text{volume}_{\text{released}})]$$

The parameters of number killed; number injured; volume released; and their associated, respective costs, namely cost per life, cost per injury, and cost of remediation are determined on a stochastic basis. In a Monte Carlo simulation, a segment along the route is selected at random and these six variables calculated. If the simulation is run enough times, the resulting means for the six variables can be used in the risk cost calculation. In this project implementation, a route segment was selected at random so that each segment had a uniform probability of being selected. In future refinements, segments in high-risk areas such as areas of high population or with cultural monuments might be given a greater weight in the selection process.

Consideration of Weights

This study in analyzing route optimality considered a third approach to risk cost calculation. In this method, different weights were assigned to each of the four significant parameters, namely haul cost, fatality, number injured, and remediation cost.

$$\text{Risk Cost}_{\text{route}} = \sum^{\text{all segments}} [w_{\text{haul cost}} * \text{cost}_{\text{car-mile}} * \# \text{cars} * \text{distance} + P(\text{attack}) * P(\text{attack occurs on segment} | \text{attack}) * [w_{\text{death}} * \# \text{killed} * \text{cost}_{\text{lives}}(\# \text{killed}) + w_{\text{injury}} * \# \text{injured} * \text{cost}_{\text{injury}}(\text{volume}_{\text{released}}) + w_{\text{remediation}} * \text{volume}_{\text{released}} * \text{cost}_{\text{remediation}}(\text{volume}_{\text{released}})]]$$

In this formulation, $w_{\text{haul cost}}$, w_{death} , w_{injury} , and $w_{\text{remediation}}$, which sum to one, correspond to degree to which a decision-maker is willing to incur a particular type of cost relative to the other costs. The $w_{\text{haul cost}}$ weight, for instance, corresponds to the degree to which the decision-maker is willing to accept the basic cost of transport relative to the other three parameters. The weights w_{death} , w_{injury} , and $w_{\text{remediation}}$ likewise correspond to the willingness to incur death, injury, and pollution. For instance, if a decision-maker is willing to incur a spill much more than assume the risk high transport cost, death, and injury, he or she might assign remediation ($w_{\text{remediation}}$) a

weighting of 75%. If he or she is indifferent of the weights of the other parameters, he or she would assign each of the remaining three weights 5% so that the sum of all the weights is 100%.

This methodology of incorporating decision-maker preferences, however, is specious because the relative value of the haul cost, death, injury, and remediation parameters are each implicit in their respective costs. Even if such weights could be incorporated, the railroads themselves would not divulge their relative preferences for incurring loss. Thus, such weights could not be determined without exhaustive research into the business practices of each railroad that operates over the route of interest.

V. RESULTS

Computational Results

The probability of attack is a product of intelligence research and thus is beyond the scope of this project. For the purposes of analyzing the model, however, a nominal, 90% confidence interval for attack probability was constructed, based on analysis of what a decision-maker would estimate with intelligence support. The TSA Railroad Security Task Force Threat Levels estimate the probability of attack on a national rail network level while this model concerns itself with the probability of attack on a per-shipment basis. Furthermore, an exponential rather than linear decrease of attack probability with an associated decrease in threat level was assumed. Therefore, the following confidence levels in Table 2 were used.

TSA Security Task Force Level			
P(attack)	4	3	2
95th %ile	0.01	0.001	0.0001
mean	0.001	0.0001	0.00001
5th %ile	0.0001	0.00001	0.000001

Table 2. Probability of Attack by Threat Level

Further inspection shows that for a given threat level, a decision-maker with intelligence support is assumed to have a confidence interval with an exponential expectation.

Based on sponsor research, a decision-maker would want to know clearly (1) the minimum risk route, (2) the maximum damage transport over a route might incur, (3) the factor that contributed most significantly to cost, and (4) the sensitivity of an optimal route to the cost per fatality. The decision-maker dashboard in Table 3 depicts this information.

Railroad Security Task Force Level 4 (Red)					
Number of Cars: 5					
	Risk Point Estimate	Risk Range	Greatest Contribution to Cost	Avg Fatality if Attack Occurs	Route 3 optimal for cost of life > \$228,958; otherwise Route 0 optimal
Route 3	\$692,497	(\$104,004 - \$6,577,422)	Loss of Life	165	
Route 2	\$770,605	(\$111,792 - \$7,358,730)	Loss of Life	185	
Route 1	\$982,871	(\$134,357 - \$9,468,010)	Loss of Life	238	
Route 0	\$1,100,930	(\$122,157 - \$10,888,660)	Loss of Life	275	

Table 3. Decision-maker Dashboard (Railroad Security Task Force Level 4)

Table 3 reports our results for Railroad Security Task Force Level 4 and five cars of chlorine. It shows that at Security Level 4, Route 3 (highlighted in yellow) is the optimal route for this threat level scenario. Route 3 has the minimum mean risk of \$692,497 and associated 90% confidence interval of (\$104,004 - \$6,577,422). Of the constituent factors of haul cost, fatality, injury, and remediation, the greatest contribution to the risk cost is the cost of lives lost.

The graph in Figure 9 shows the sensitivity analysis of the cost of life at Security Level 4 to determine the breakpoint number. The optimal route for a given cost of life is the one with the minimum risk cost, which is displayed on the y-axis. Negative costs per life are not considered. Route 3 maintains its optimality for all costs per life greater than \$228,958; otherwise, Route 0 becomes optimal. The cost per life breakpoint of \$228,958 corresponds to a risk cost of \$76,437 for Routes 0 and 3.

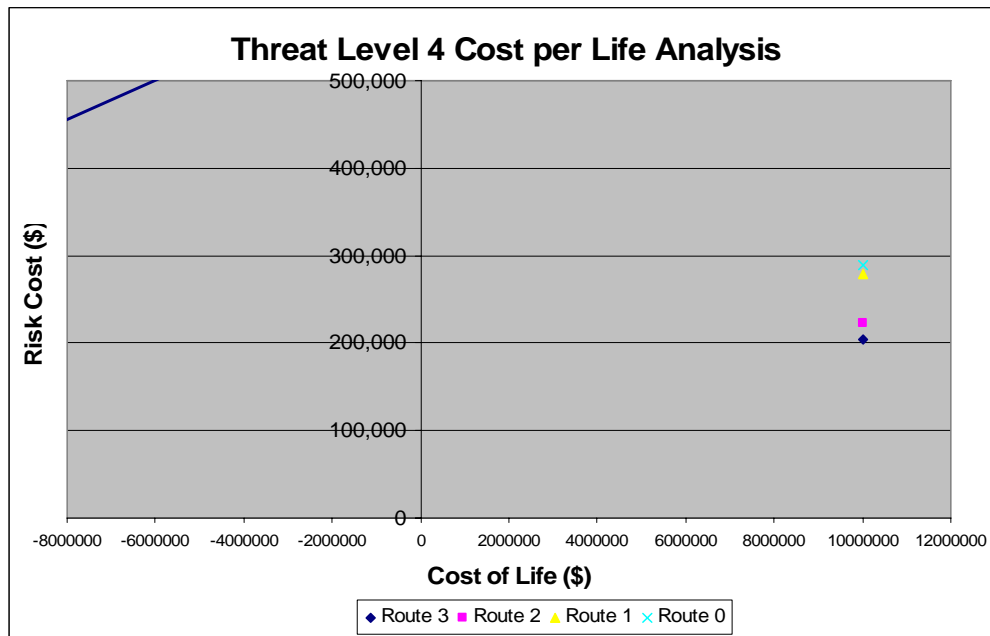


Figure 9. Cost of Life Analysis (Threat Level 4)

Railroad Security Task Force Level 3 (Orange)					
Number of Cars: 5					
	Risk Point Estimate	Risk Range	Greatest Contribution to Cost	Avg Fatality if Attack Occurs	Route 3 optimal for cost of life > \$2,291,576; otherwise Route 0 optimal
Route 3	\$104,004	(\$45,155 - \$692,497)	Loss of Life	165	
Route 2	\$111,792	(\$45,911 - \$770,605)	Loss of Life	185	
Route 0	\$122,157	(\$24,280 - \$1,100,930)	Loss of Life	275	
Route 1	\$134,357	(\$49,506 - \$982,871)	Loss of Life	238	

Table 4. Decision-maker Dashboard (Railroad Security Task Force Level 3)

At Railroad Security Task Force Level 3 with five cars of chlorine, Route 3 has the lowest mean risk, depicted by the risk point estimate, and is once again considered to be the optimal route. Note that route 3’s lower 90% confidence interval bound, \$45,155, is higher than that of Route 0, \$24,280; however, route 3’s expected risk, \$104,004, and upper 90% confidence interval bound, \$692,497, are lower than those of route 0, which are \$122,157 and \$1,100,930, respectively. Since decision-makers tend to adopt a not-on-my-watch approach to risk management, the mean expected risk and the upper bound together are considered more important than the lower 90% confidence interval bound. For Security Level 3, the greatest contribution to cost remains the cost of life. Route 3 maintains its optimality for all costs per life greater than \$2,291,576; otherwise, Route 0 becomes optimal. At this breakpoint, the risk cost of routes 0 and 3 is \$76434. The sensitivity analysis performed to find this breakpoint is illustrated in Figure 10.

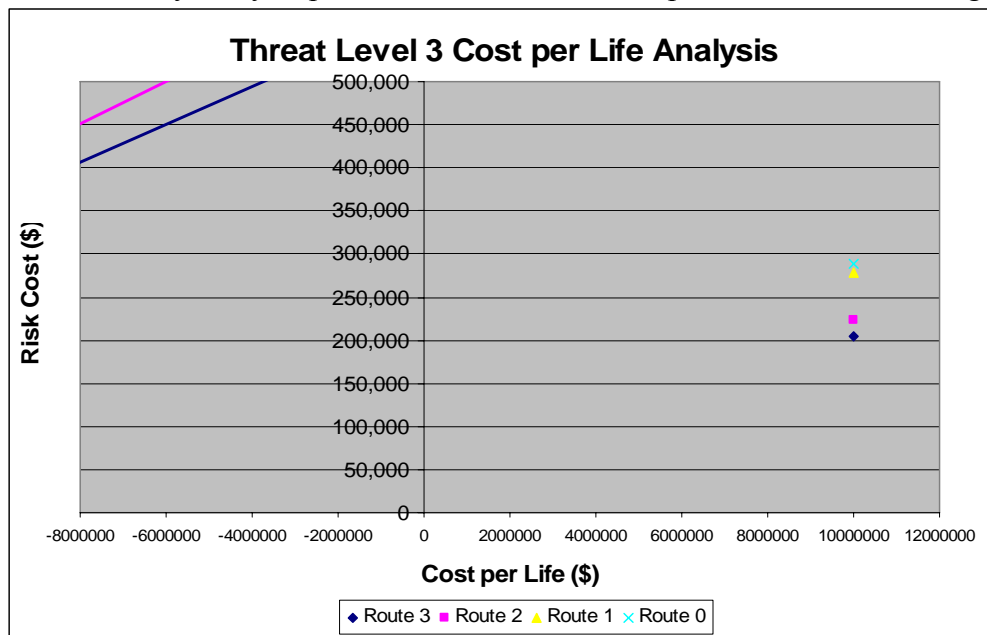


Figure 10. Cost of Life Analysis (Threat Level 3)

Railroad Security Task Force Level 2 (Yellow)					
Number of Cars: 5					
	Risk Point Estimate	Risk Range	Greatest Contribution to Cost	Avg Fatality if Attack Occurs	Route 3 optimal for cost of life > \$22,915,998; otherwise Route 0 optimal
Route 0	\$24,280	(\$14,492 - \$122,157)	Haul Cost	275	
Route 3	\$45,155	(\$39,270 - \$104,004)	Haul Cost	165	
Route 2	\$45,911	(\$39,323 - \$111,792)	Haul Cost	185	
Route 1	\$49,506	(\$41,021 - \$134,357)	Haul Cost	238	

Table 5. Decision-maker Dashboard (Railroad Security Task Force Level 2)

At Level 2, there are two optimal routes. Route 0 has the lowest mean risk cost, \$24,280, and lowest 90% confidence interval lower bound, \$14,492, but Route 3 has a lowest cost for the 95% confidence interval upper bound, \$104,004. The decision-maker must weigh to what degree he or she feels comfortable with the increased upper bound on the risk that Route 0 provides. At this point, the decision-maker may want to look at the maximum damage table to help decide which route to use.

	Maximum Damage
Route 0	\$1,333,696,333
Route 1	\$1,333,723,006
Route 2	\$1,333,721,519
Route 3	\$1,333,721,544

Table 6. Maximum Damage per Route

Since the damages are approximately the same for Route 0 and Route 3, and in fact lower for Route 0, the decision-maker may be influenced to select Route 0 with the lower mean risk cost.

In performing sensitivity analysis of the cost per life to determine the cost breakpoint, Route 3 is optimal for costs per life greater than \$22,915,998; otherwise Route 0 optimal. As with Threat Level 3, the risk cost of routes 0 and 3 is \$76,434 at this breakpoint. All three Threat Levels have a cost per life breakpoint for Routes 0 and 3. The variance of the Level 2 and 3 risk cost at the breakpoint with that of the Threat Level 4 breakpoint, namely \$76,437, can be attributed to rounding error. Thus, the cost per life breakpoint always occurs at the same risk cost, approximately \$76,435. Figure 11 depicts the Threat Level 2 breakpoint.

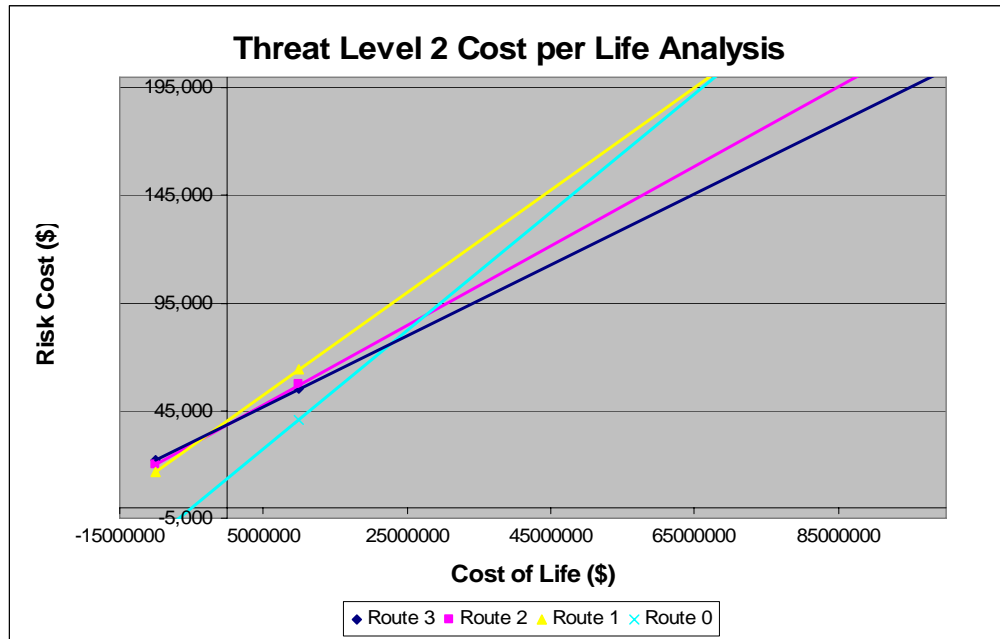


Figure 11. Cost of Life Analysis (Threat Level 2)

Once the decision-maker has selected a security level of interest and a route, he or she may wish to know the optimal number of cars to transport along that route. A drill down mechanism in the dashboard enables the decision-maker to look at the risk cost according to the 5 per cent, 50 per cent, and 95 per cent confidence level for attack probability as shown in the graph in Figure 12.

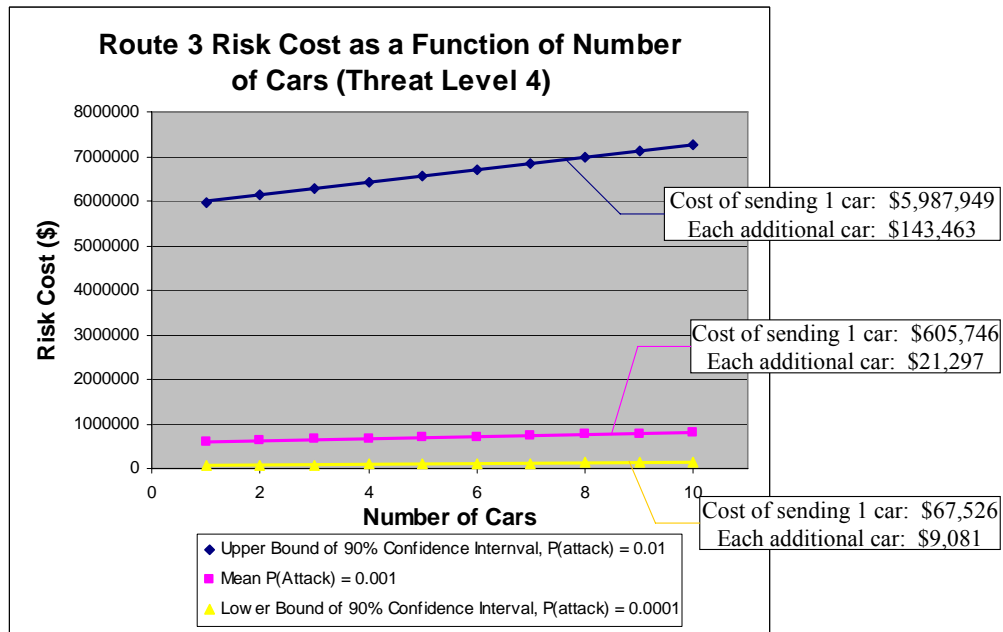


Figure 12. Optimum Number of Cars to Ship along Route 3

In this example, we see that for all probabilities of attack within the 90% confidence interval, the risk cost of the initial investment in sending a shipment outweighs the risk cost of adding

additional tank cars. Thus once the decision-maker decides to send a shipment, he or she should send the maximum number of chlorine cars that will be purchased.

Stochastic Model Results:

As discussed in the previous optimization section, an alternative formulation of the model was implemented using a uniform distribution to select the route segment under attack as opposed to a selection based on population density ratio as done with the deterministic approach described above. The results of this stochastic approach are described in Table 7.

Railroad Security Task Force Level 4 (Red)					
Number of Cars: 5					
	Risk Point Estimate	Risk Range	Greatest Contribution to Cost	Avg Fatality if Attack Occurs	Route 3 Optimal for all Values per Life
Route 3	\$189,943	(\$53,749 - \$1,551,887)	Cost of Life	38	
Route 2	\$205,014	(\$55,233 - \$1,702,821)	Cost of Life	41	
Route 1	\$281,099	(\$64,180 - \$2,450,291)	Cost of Life	60	
Route 0	\$798,907	(\$91,955 - \$7,868,427)	Cost of Life	198	

Table 7. Decision-maker Dashboard (Railroad Security Task Force Level 4)

Table 7 reports our results for Railroad Security Task Force Level 4 and five cars of chlorine. It shows that at Security Level 4, Route 3 (highlighted in yellow) is the optimal route for this threat level scenario. Route 3 has the minimum mean risk of \$189,943 and associated 90% confidence interval of (\$53,749 - \$1,551,887). Of the constituent factors of haul cost, fatality, injury, and remediation, the greatest contribution to the risk cost is the cost of lives lost.

Railroad Security Task Force Level 3 (Orange)					
Number of Cars: 5					
	Risk Point Estimate	Risk Range	Greatest Contribution to Cost	Avg Fatality if Attack Occurs	Route 3 Optimal for all Values per Life
Route 3	\$53,749	(\$40,130 - \$189,943)	Haul Cost	38	
Route 2	\$55,233	(\$40,255 - \$205,014)	Haul Cost	41	
Route 1	\$64,180	(\$42,488 - \$281,099)	Haul Cost	60	
Route 0	\$91,955	(\$21,259 - \$798,907)	Cost of Life	198	

Table 8. Decision-maker Dashboard (Railroad Security Task Force Level 3)

At Security Level 3, Route 3 has the lowest mean risk, depicted by the risk point estimate, and is once again considered to be the optimal route. This time, however, the greatest contribution to the risk cost was the haul cost.

Railroad Security Task Force Level 2 (Yellow)					
Number of Cars: 5					
	Risk Point Estimate	Risk Range	Greatest Contribution to Cost	Avg Fatality if Attack Occurs	Route 0 Point Estimate Optimal for Values per Life Less than \$14,874,468
Route 0	\$21,259	(\$14,190 - \$91,955)	Haul Cost	198	
Route 3	\$40,130	(\$38,768 - \$53,749)	Haul Cost	38	
Route 2	\$40,255	(\$38,757 - \$55,233)	Haul Cost	41	
Route 1	\$42,488	(\$40,319 - \$64,180)	Haul Cost	60	

Table 9. Decision-maker Dashboard (Railroad Security Task Force Level 2)

At Level 2, there are two optimal routes. Route 0 has the lowest mean risk cost, but Route 3 has a lower cost for the 95% confidence interval upper bound. The maximum damages are depicted in Table 6 and so the decision-maker may select Route 0, which has the lower mean risk cost and maximum damage value.

Previous discussions on risk estimation can now be used to populate the generic decision model presented in Section 3. Figure 13 shows the complete set of data used for the decision tree calculations that include TSA warning levels (with an arbitrary distribution of days of the year at a specific threat level), risk function output data, and the EMV comparison.

The decision tree shows that if the railroad were to re-route TIH material transport trains as a function of increasing TSA warning levels, they would decrease their risk by 9% for this specific route selection and threat chances (see Figure 14). The annual shipment rate is assumed to be 8500 tank cars per year and 5 tank cars per shipment.

Based on this initial result, the recommendation for the decision-maker is to further explore the option of following the TSA countermeasure model because this course of action satisfies the railroad company’s objective of reducing TIH train en-route security risk. This may involve access to proprietary financial and operational data owned by the railroads and intelligence community organizations.

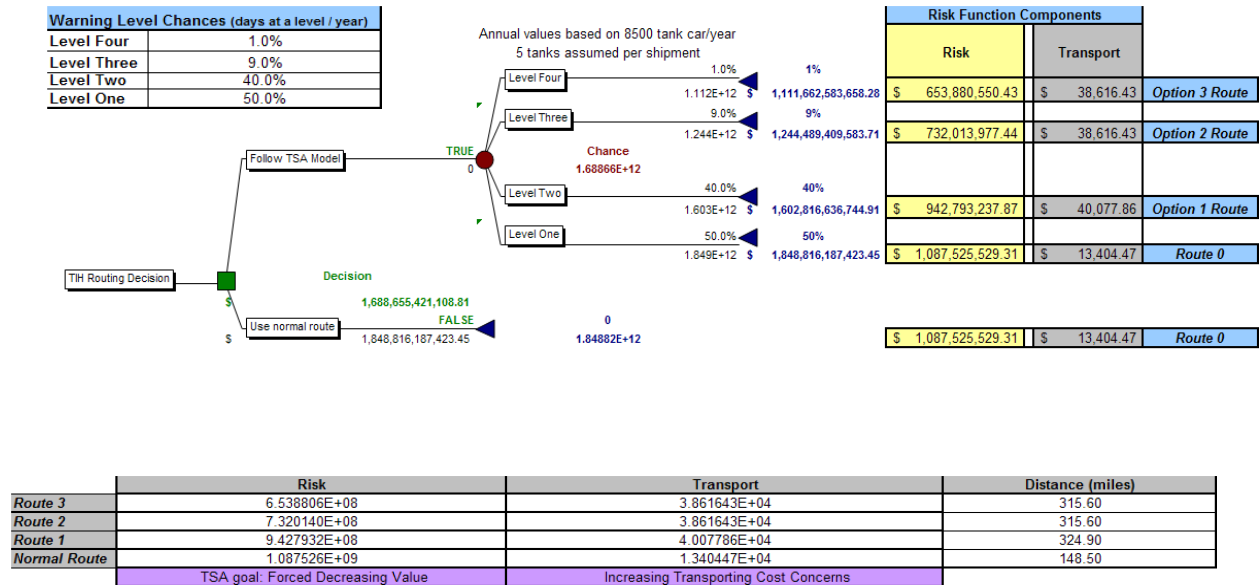


Figure 13. Decision Tree and Analysis Framework

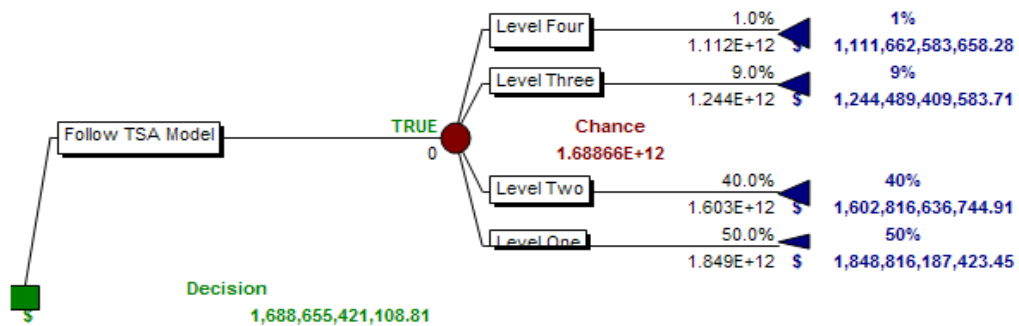


Figure 14. Decision analysis output: policy suggestion and recommendations

VI. CONCLUSIONS

As a result of this effort, findings in three different areas were obtained: observations are made based on experience gained as the project progressed; recommendations are made concerning potential future work; and conclusions have been derived that are supported by the results of modeling and analysis.

In the area of observations, it was learned that the proprietary nature of the railroad industry's operational and financial data greatly limits development of comprehensive decision analysis capabilities to advance the sponsor's goals for decision support to operational risk management. For example, information on shipment costs of TIH material was very difficult to obtain; furthermore, costs pertaining to mitigation of uncontrolled TIH material release events are very closely held by legal agreement. Consequently, in the timeframe of this project, it was not possible to obtain access to much of the data needed to make well-informed assessments. However, assumptions can be made based on limited publicly available data to substitute for missing values. It appears these are valid in terms of railroad industry norms. It was also learned that there are several railroad community stakeholders with conflicting objectives. For example, operational traffic planners in the railroad industry seek to minimize risk but from the standpoint of what is the impact of risk exposure on the company's revenue base. But a second group of stakeholders has a supervisory role with respect to regulatory compliance; specifically the FRA. A third group of stakeholders, the TSA Railroad Security Task Force, is concerned solely with transportation security assessments and advisory functions needed for threat response management. Risk management is highly influenced by decision-maker preferences and requires agreement of the multiple stakeholders that participate in the process. It may be challenging to obtain sufficient consensus between such stakeholder groups to facilitate consensus in this community. It was also learned that the railroad industry is highly regulated. The regulatory environment appears to be driving the realities of risk management in terms of TIH material transport. For example, there is some opinion in the RR industry that the risk of TIH transport is too high to justify its transport; that is, the potential revenue is not worth the risk of transport at any market price. However, as regulated common carriers, the RR industry is required to transport TIH materials. Consequently, the RR industry is forced to assume risk that in some cases would not be evaluated as profitable and would otherwise be deemed unreasonable to bear. For example, risk exposure in cost terms could be in the billions of dollars in some event scenarios. Insurance can be purchased but the potential magnitude of any TIH material transport incident occurring on a catastrophic scale could far exceed available limits. Some companies in the Class I railroad industry tier describe TIH material transportation as a bet-the-company risk.

Observations can also be made concerning how the RR team managed itself. Model development was an iterative process. In each iteration, the team analyzed the data output and reviewed and revised the model. By implementing several different approaches and responding to sponsor needs and feedback the team deviated from its initial plan of work. This was necessary to be able to undertake new approaches as collective understanding evolved. Extended effort was also required to account for multiple different approaches. This evolutionary process is reflective of an exploratory project where requirements and techniques are developed through multiple steps in a process of refinement. The plan of work could have been updated as each

step became clear, but a better method would have been to devise and adopt a work plan based on applying fixed levels of effort corresponding to specific products, rather than making estimates of how many staff-hours are required to complete each step to produce each deliverable.

Recommendations pertaining to future work can also be made. Primarily these concern functional analyses and the technical approach. In the area of functional analysis, it could be instructive to consider more complex route origin-destination pairs than time allowed in this effort. If there were more than two possible routes in the set of possible operationally viable routes, other decision factors could emerge. Another recommendation could be made to consider evaluation of security risk as a function of the number of people exposed, rather than the cost consequence of exposing those people. In the technical approach, due to the limited timeframe for this project, it was not possible to further explore how to model and possibly resolve the effects on risk of ambiguity in the threat definition. Due to imperfect threat information, there will always be a level of ambiguity in threat characterization. Consequently, future work should possibly consider stochastic techniques to evaluate which segments of a route might be attacked among a set of approximately equally ranked candidates. The set of equal candidates could be chosen using a value assessment method, assuming an intelligent or rational threat actor, but which segment in that set might be attacked could be studied as a choice made at random, reflecting the ambiguity typically inherent to threat characterization. This approach could allow identification and analysis of a smaller set of candidate segments. It could also facilitate emergence of other decision factors.

The principal conclusion of this effort is as follows: the model results support the problem statement, specifically that a cost-based risk assessment tool can enable decision-makers to be better informed about TIH material transport route selection, while incorporating analysis of costs, alternative routes, and simplification of results.

Within the scope of the limited timeframe for this project, results to date show that the model supports cost-based risk analysis and assessments of alternative routes for TIH material transport. Factors include cost of transport (haul cost), number killed, number injured, and cost of remediation. Model results show that certain routes are higher cost in terms of consequences than others. Results show that if the probability of attack is sufficiently low, such as TSA Security Level 2, then haul cost is the driving factor because consequences due to fatalities are significantly less. However, for TSA Security Levels 3 and 4, then the consequence costs due to fatalities are significantly higher, because the probability of attack is the fundamental driving factor. At high probabilities of attack, that is, TSA Security Levels 3 and 4, the model results show that the decision-maker should choose routes that are inherently less costly in terms of risk exposure. Those are the routes with the risk of fewer fatalities.

An example of the trade-off analyses that can be performed based on model results is seen in the case where at TSA Security Level 3, Route 3 has the lowest mean risk, but Route 3's lower 90% confidence interval bound, \$45,155, is higher than that of Route 0, \$24,280; however, Route 3's expected risk, \$104,004, and upper 90 per cent confidence interval bound, \$692,497, are lower than those of Route 0, which are \$122,157 and \$1,100,930, respectively (see Table 4). Decision-makers can make choices based on such considerations. If they are willing to gamble on the

prospect that Route 0's risk at the lowest range is less than Route 3, then they could choose Route 0. The decision-maker can subsequently conduct trade-offs on the number of TIH material transport cars to be sent on a specific route.

Another example of trade-off analysis that can be performed based on model results is seen in the case where at TSA Security Level 4, Route 3 maintains its optimality for all costs per life greater than \$228,958 (see Table 3). Below this cost per life, Route 0 is optimal. It is possible to conduct trade-offs on alternative routes based on expected cost per life. This could be borne out operationally in cases where these are locales where settlements to such incidents traditionally have been lower than in others; for example, in rural vice urban settings. While such trade-offs based on cost of life may not seem appropriate, it is a fact of the body of law relevant to such cases.

Another example of trade-off analysis that can be performed based on model results is seen in the case where at TSA Security Level 2, there are two optimal routes. Route 0 has the lowest mean risk cost, \$24,280, and lowest 90 per cent confidence interval lower bound, \$14,492, but Route 3 has a lowest cost for the 95 per cent confidence interval upper bound, \$104,004 (see Table 5). The decision-maker can make choices based on to what degree they feel comfortable with the increased upper bound on the risk that Route 0 provides.

The main contribution or impact of this effort has been to demonstrate that cost-based risk analysis is a viable analytical model for assessment of alternative TIH material transport routes. This can be seen from the model results, which show sensitivity of route selection to cost as a risk assessment variable. This effort has also shown the potential for future work in use of cost-based risk assessment techniques for TIH material transport.

Obstacles to implementation of the proposed solution include a need to conduct further analyses of operational needs, for example ambiguities in the threat information. Recommendations for follow-on work in this area have been made.

Given the timeframe for this project, the sponsors are satisfied with the team's interaction with them and with the outcome. For some routes, en-route security risk reduction can be nearly 50 per cent from normative route risk. Without a cost-based risk assessment tool, the stakeholders would have little other than intuitive knowledge of such cost exposure. The proposed decision analysis framework can help the stakeholders in understanding the problem, harmonizing conflicting objectives and reaching a sufficient consensus to support informed decision-making.

Appendix A: Operational Requirements

1. TIH material transport compatible routes shall be selected based on several parameters:
 - a. The wayside defective bearing detectors shall be placed at a maximum of 40 miles apart
 - b. Key Routes shall be defined as any track with a combination of 10,000 car loads or intermodal portable tank loads of hazardous materials, or a combination of 4,000 car loadings of PIH or TIH materials (Hazard zone A, B, C, or D), anhydrous ammonia, flammable gas, Class 1.1 or 1.2 explosives, environmentally sensitive chemicals, Spent Nuclear Fuel (SNF), and High Level Radioactive Waste (HLRW) over a period of one year.
 - c. Key Trains shall be defined as a train with: Five tank car loads of Poison or Toxic Inhalation Hazard1 (PIH or TIH) (Hazard Zone A, B, C, or D)
 - d. Key Trains shall hold main track at meeting or passing points, unless siding or auxiliary track meets FRA Class 2 standards
 - e. The maximum reasonable effort shall be made to achieve coupling of loaded placarded tank cars at speeds not to exceed 4 MPH.
 - f. The main track on "Key Routes" shall be inspected by rail defect detection and track geometry inspection cars or any equivalent level of inspection no less than two times each year;
 - g. The rail sidings shall be inspected no less than one time each year.
 - h. The main track and sidings shall have periodic track inspections that will identify cracks or breaks in joint bars.
2. Threat attacks shall be minimized based on:
 - i. Population
 - j. Route length
 - k. Economic impacts
 - l. TIH material exposure and costs
3. The en-route train risk shall be defined by:
 - a. Minimize TIH material Rail Route Risk
 - b. Minimize Probability
 - Minimize shipments of TIH materials
 - c. Minimize Vulnerability
 - Minimize Safety
 - Track infrastructure
 - Tank survivability
 - Maximize Security
 - TIH material car dwelling time
 - TSA threats
 - d. Minimize Consequences
 - Minimize Public Damage
 - Loss of life
 - Injury
 - Minimize Cost of Excess Costs
 - Re-routing shipper's cost
4. Optimization Requirements:

- a. The optimization model shall be contain three attributes to represent the most important factors for the decision-maker's route selection for each route
 - i. Cost of Sending Freight (cost/ton-mile)
 - ii. Volume of TIH material per train
 - iii. Population Densities (PD)
- b. The common units of measure for the model shall be determined for each attribute:
 - i. Cost/ton-mile
 - ii. Volume of TIH material
- c. The routes for the model shall be optimized based on metrics:
 - i. Incorporate Security Risk (Threat) Factors
 - i. Determine Probability Functions
 - a. Probability of Attack
 - b. Probability train car of CL2 ruptures if there is a hit
 - c. Probability of TIH material exposure
- d. The cost factors for the model shall be determined by developing a cost function:
 - i. Cost of Lives
 - d. Dependent on number killed
 - ii. Cost of Injury
 - e. Dependent on numbered injured
- e. A software program shall be created to produce data results
 - i. Route Costs
 - ii. Exposure Costs
- f. A Sensitivity Analyses shall be performed on the model results
 - i. Provide algorithm implementation options
- g. The RRVA and TRAGIS tools shall be utilized for the development of the Railroad Team's optimization model

Appendix B: Project Management

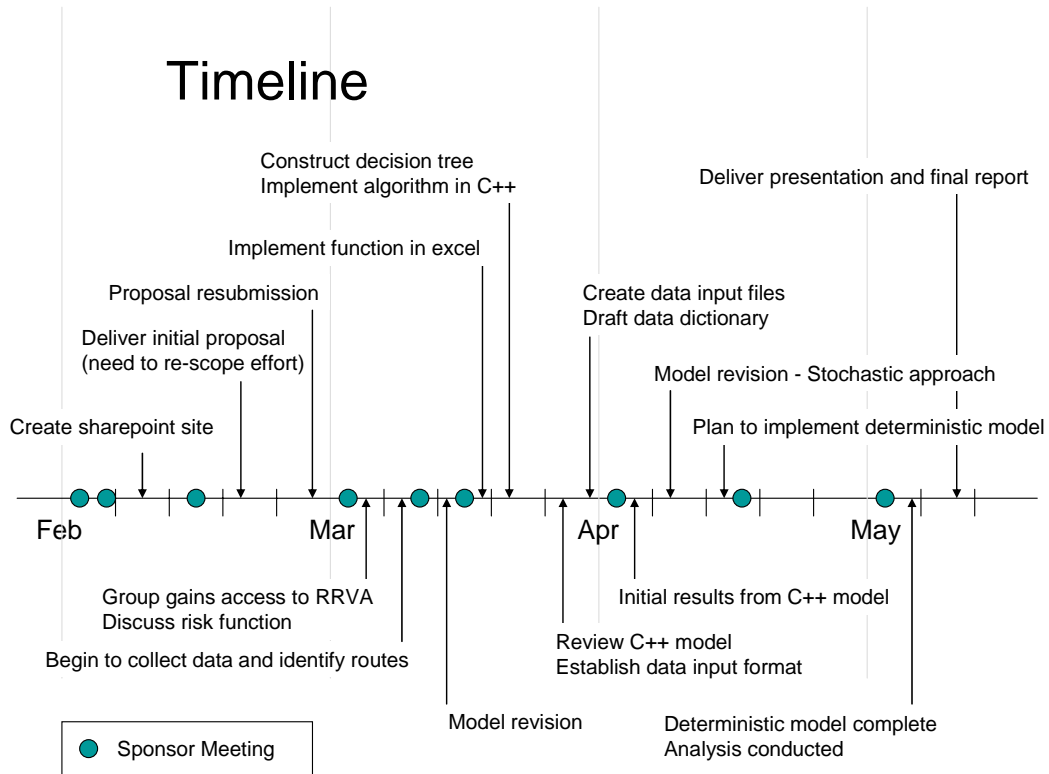


Figure 15. Project Timeline

The timeline of the team’s effort shown in Figure 15 illustrates the major turning points in our work. Model development was an iterative process. In each iteration, our team analyzed the data output and reviewed and revised the model. The initial implementation was done in Excel, however we were unable to perform all of the analyses originally planned using this software. The model was then implemented in C++ using Borland’s C++ Builder VI interactive development environment. The model summed costs over all segments and tested a variety of weighting schemes solicited from the sponsors acting as subject matter experts (SMEs). This model resulted in especially large risk costs due to the fact that it assumed an accident would occur at each segment along the route. After each project status briefing, we addressed questions and concerns identified by Dr. Laskey, fellow students and our sponsors. In response a stochastic model was developed which selected a route segment at random and calculated risk cost using different values for probability of attack to represent the TSA Railroad Security Task Force threat levels. After further review it was clear that a rational-attacker method for segment selection should be modeled. A deterministic approach was taken which assumes that an attacker would seek to inflict maximum damage by attacking an optimal segment. Since we are investigating this problem to protect against terrorist threat, segments are chosen to maximize the population exposed.

By implementing several different approaches and responding to sponsor needs and feedback the team deviated from the plan of work. The team undertook several approaches to address the problem as collective understanding evolved. This extended effort accounts for the additional person-hours depicted in the earned value management graph in Figure 16. However this evolutionary process is reflective of an exploratory project where requirements and techniques

are developed through multiple steps in a process of refinement. The team could have updated the plan of work shown in Figure 17 as each step became clear, but a better technique could have been to define a work plan that is based on level of effort corresponding to specific products, rather than person-hour estimates required to produce each deliverable.



Figure 16. Earned Value Management

Week Complete	TASK	Hours Allocated
all weeks	Project Management Framework	403
	Initiating	
1	Develop Project Scope and Goals	25
2	Define deliverables and expected results	25
	Planning	
3	Identify constraints & assumptions	24
3	Define Project Components and Stakeholders	24
3,4	Generate Cost/Resource/Tasking	10
	Executing	
3	Maintain traceability to sponsor requirements	15
all weeks	Attend internal and sponsor meetings	35
	Monitor and Control	
all weeks	Monitor Technical performance	40
all weeks	Monitor and Control schedule	40
	Closure	
1	Project Plan and Proposal	25
2	Status Report Deliverable 1	15
3	Progress Report	15
5	Status Report Deliverable 2	10
7	Team Progress Review Meeting	25
9	Formal Progress Presentation	25
11	Formal Dry-Run Presentation	25
13	Final Presentation and Final Report due	25
Weeks 3-8	SE and OR Development	176
	Problem Identification	
4	Identify approach	25
5	Identify and classify risks	30
	Perform Research	
3,4,5	Literature Review	50
5	Obtain railroad data,statistics and routes	21
	Develop Product	
5	Define Optimization Model parameters	5
5	Identify all assumptions	4
5,6	Identifies variables, weights, constants, and metrics	5
6	Identify scenarios and calculate algorithm	6
	Perform Sensitivity Analysis	
7,8	Perform analysis for various sets of probabilities	30
Weeks 8-10	Integration and Testing	85
	Integrate	
8	Algorithm Integration - RRVA	25
9,10	Compare algorithm output	15
	Test and Evaluate	
9,10	Test Evaluation parameters	15
9,10	Evaluate weaknesses and strengths of RRVA and optimization algorithm.	30
Weeks 10-12	Operations and Maintenance	32
	Web services	
10	Maintain Web-site for Project	20
	Manufacture Product for RRVA	
11	Manufacture Algorithm for RRVA	9
12	Produce Lessons Learned	3
	Totals:	696

Figure 17. Work Plan

Appendix C: Deterministic Model

```
// Deterministic Risk Cost program
// Authors: Matthew Albin, Ruben Luna, Danielle Martin, Danielle Obuchon, Andy
// Ramsay
// 3 May 2008
```

```
#include <iostream.h>
#include <math.h>
#include <fstream.h> // file I/O
#include <iomanip.h> // format manipulation
```

```
class Route {
    int routeNum; //unique route number
    int numSegs; //number of segments in the route--max is 50 (size of
                //array below)
    int segNums[100]; //segment numbers constituting route

public: void set_values (int in_routeNum, int in_numSegs, int in_segNums[])
    {
        routeNum = in_routeNum;
        numSegs = in_numSegs;
        for (int i = 0; i < numSegs; i++)
        {
            segNums[i] = in_segNums[i];
        }
    }
    void print_values (void)
    {
        cout << "Route " << routeNum << " contains " << numSegs <<
            " segments, segments ";
        int i, currentSeg;
        for (i = 0; i < numSegs-1; i++)
        {
            currentSeg = getSegNum(i);
            cout << currentSeg << ", ";
        }
        currentSeg = getSegNum(i);
        cout << currentSeg << "." << endl;
    }
    int getRouteNum (void) {return routeNum;}
    int getNumSegs (void) {return numSegs;}
    int getSegNum (int segNumCounter) {return segNums[segNumCounter];}
};
```

```
class Segment {
```

```

int startNode;
int endNode;
double cost_car_mile;
double PV;
double pop_seg;
double dist_miles;
double dist_meters;
int carrier;

public:
void set_values (int in_startNode, int in_endNode, double in_cost_car_mile,
                double in_PV, double in_pop_seg, double in_dist_miles,
                int in_carrier)
{
    startNode = in_startNode;
    endNode = in_endNode;
    cost_car_mile = in_cost_car_mile;
    PV = in_PV;
    pop_seg = in_pop_seg;
    dist_miles = in_dist_miles;
    dist_meters = dist_miles * 1609.344;
    carrier = in_carrier;
}
void print_values (void)
{
    cout << " Start node is " << startNode << "." << endl;
    cout << " End node is " << endNode << "." << endl;
    cout << " Cost per car-mile is " << cost_car_mile << endl;
    cout << " # Points of vulnerability is " << PV << endl;
    cout << " Population along segment is " << pop_seg << endl;
    cout << " Distance in miles is " << dist_miles << endl;
    cout << " Distance in meters is " << dist_meters << endl;
    cout << " Carrier is ";
    if (carrier == 0) cout << "Norfolk Southern";
    else if (carrier == 1) cout << "CPRS";
    else if (carrier == 2) cout << "CSX";
    else if (carrier == 3) cout << "CR";
    else cout << "unknown";
    cout << "." << endl;
}
double getCost_CarMile () {return cost_car_mile;}
double getPV () {return PV;}
double getPopSeg () {return pop_seg;}
double getDistMiles () {return dist_miles;}
double getDistMeters () {return dist_meters;}
};

```

```
void main()
{
  int sillyVariableToKeepTheProgramWindowOpen;
  ofstream fp_out;
  fp_out.open("determ_version2.txt", ios::out);
  fp_out << setprecision(14);

  Segment *Segment_Array;
  Segment_Array = new Segment[10000];

  Segment_Array[0].set_values(512113015,512113023,24.77,0,8450,0,0);
  Segment_Array[1].set_values(512113023,512113027,24.77,0,2450,1.7,0);
  Segment_Array[2].set_values(512113027,512113038,24.77,0,2450,2.4,0);
  Segment_Array[3].set_values(512113038,512113037,24.77,0,2450,13.8,0);
  Segment_Array[4].set_values(512113037,512113054,24.77,0,3530,0.5,0);
  Segment_Array[5].set_values(512113054,512113058,24.77,0,3530,1.4,0);
  Segment_Array[6].set_values(512113058,512113075,24.77,0,4130,0.2,0);
  Segment_Array[7].set_values(512113075,512113073,24.77,0,4130,2.1,0);
  Segment_Array[8].set_values(512113073,512113062,24.77,0,4130,0.3,0);
  Segment_Array[9].set_values(512113062,512113000,24.77,0,85,1.1,0);
  Segment_Array[10].set_values(512113000,512112885,24.77,0,85,12.7,0);
  Segment_Array[11].set_values(512112885,512112850,24.77,0,85,28.2,0);
  Segment_Array[12].set_values(512112850,512112848,24.77,0,85,7.2,0);
  Segment_Array[13].set_values(512112848,512112839,24.77,0,85,0.9,0);
  Segment_Array[15].set_values(512112836,512112746,24.77,0,140,0.1,0);
  Segment_Array[16].set_values(512112746,522112389,24.77,0,140,7,0);
  Segment_Array[17].set_values(522112389,542112172,24.77,0,240,16.6,0);
  Segment_Array[18].set_values(542112172,542112069,24.77,0,240,8,0);
  Segment_Array[19].set_values(542112069,522111970,24.77,0,290,4.1,0);
  Segment_Array[20].set_values(522111970,242111643,24.77,0,290,6.6,0);
  Segment_Array[21].set_values(242111643,242111635,24.77,0,290,15.3,0);
  Segment_Array[22].set_values(242111635,522111481,24.77,0,290,0.4,0);
  Segment_Array[23].set_values(522111481,422110664,24.77,0,290,6.5,0);
  Segment_Array[24].set_values(422110664,422110660,24.77,0,400,26.3,0);
  Segment_Array[25].set_values(22110660,422110649,24.77,0,400,0.1,0);
  Segment_Array[26].set_values(422110649,422110515,24.77,0,400,0.6,0);
  Segment_Array[27].set_values(422110515,422110471,24.77,0,400,19,0);
  Segment_Array[28].set_values(422110471,422110454,24.77,0,400,8.7,0);
  Segment_Array[29].set_values(422110454,422110360,24.77,0,400,2.2,0);
  Segment_Array[30].set_values(422110360,422110296,24.77,0,400,4.4,0);
  Segment_Array[31].set_values(422110296,422110267,24.77,0,400,3,0);
  Segment_Array[32].set_values(422110267,422110256,24.77,0,400,1.8,0);
  Segment_Array[33].set_values(422110256,422110254,24.77,0,480,0.8,0);
  Segment_Array[34].set_values(422110254,422110258,24.77,0,480,0.2,0);
```

```
Segment_Array[35].set_values(422110258,422110257,24.77,0,480,0.3,0);
Segment_Array[36].set_values(422110257,422110227,24.77,0,480,0.5,0);
Segment_Array[37].set_values(422110227,422110228,24.77,0,480,7,0);
Segment_Array[38].set_values(422110228,422110230,24.77,0,480,0.6,0);
Segment_Array[39].set_values(422110230,422110150,24.77,0,480,0.2,0);
Segment_Array[40].set_values(422110150,422110059,24.77,0,335,6.3,0);
Segment_Array[41].set_values(422110059,422110058,24.77,0,335,9.5,0);
Segment_Array[42].set_values(422110058,422110029,24.77,0,335,0.2,0);
Segment_Array[43].set_values(422110029,422110100,24.77,0,435,12.9,0);
Segment_Array[44].set_values(422110100,422110072,24.77,0,435,9.5,0);
Segment_Array[45].set_values(422110072,422110068,24.77,0,435,3.6,0);
Segment_Array[46].set_values(422110068,422110066,24.77,0,435,0.5,0);
Segment_Array[47].set_values(422110066,422110153,24.77,0,435,0.2,0);
Segment_Array[48].set_values(422110153,422110124,24.77,0,435,3.5,0);
Segment_Array[49].set_values(422110124,422110219,24.77,0,435,0.9,0);
Segment_Array[50].set_values(422110219,422110221,24.77,0,435,6.2,0);
Segment_Array[51].set_values(422110221,422110270,24.77,0,435,0.2,0);
Segment_Array[52].set_values(422110270,422110271,24.77,0,552,6.7,0);
Segment_Array[53].set_values(422110271,422110331,24.77,0,552,2,0);
Segment_Array[54].set_values(422110331,422110457,24.77,0,552,5,0);
Segment_Array[55].set_values(422110457,422110460,24.77,0,552,7,0);
Segment_Array[56].set_values(422110460,422110485,24.77,0,552,0,0);
Segment_Array[57].set_values(422110485,422110522,24.77,0,552,1,0);
Segment_Array[58].set_values(422110522,422110528,24.77,0,552,2.5,0);
Segment_Array[59].set_values(422110528,422110558,24.77,0,552,0.3,0);
Segment_Array[60].set_values(422110558,422110543,24.77,0,552,1.3,0);
Segment_Array[61].set_values(422110543,422110532,24.77,0,552,3.1,0);
Segment_Array[62].set_values(422110532,422110540,24.77,0,552,1.4,0);
Segment_Array[63].set_values(422110540,422110552,24.77,0,552,1.3,0);
Segment_Array[64].set_values(422110552,422110566,24.77,0,552,1.1,0);
Segment_Array[65].set_values(422110566,422110604,24.77,0,552,1,0);
Segment_Array[66].set_values(422110604,422110673,24.77,0,552,1.4,0);
Segment_Array[67].set_values(422110673,422110690,24.77,0,552,5.6,0);
Segment_Array[68].set_values(422110690,422110733,24.77,0,552,0.5,0);
Segment_Array[69].set_values(422110733,422110744,24.77,0,552,2.1,0);
Segment_Array[70].set_values(422110744,422110754,24.77,0,552,0.7,0);
Segment_Array[71].set_values(422110754,422110813,24.77,0,11230,0.2,0);
Segment_Array[72].set_values(422110813,422110931,24.77,0,11230,1.3,0);
Segment_Array[73].set_values(422110931,422110931,24.77,0,11230,3.8,0);
```

```
Segment_Array[1063].set_values(422110540,422110552,21.37,0,552,1.3,1);
Segment_Array[1064].set_values(422110552,422110566,21.37,0,552,1.1,1);
Segment_Array[1065].set_values(422110566,422110604,21.37,0,552,1,1);
Segment_Array[1066].set_values(422110604,422110673,21.37,0,552,1.4,1);
Segment_Array[1067].set_values(422110673,422110690,21.37,0,552,5.6,1);
```

```
Segment_Array[1068].set_values(422110690,422110733,21.37,0,552,0.5,1);  
Segment_Array[1069].set_values(422110733,422110744,21.37,0,552,2.1,1);  
Segment_Array[1070].set_values(422110744,422110754,21.37,0,552,0.7,1);  
Segment_Array[1071].set_values(422110754,422110813,21.37,0,11230,0.2,1);  
Segment_Array[1072].set_values(422110813,422110931,21.37,0,11230,1.3,1);  
Segment_Array[1073].set_values(422110931,422110931,21.37,0,11230,3.8,1);
```

```
Segment_Array[77].set_values(422110931,422110943,24.77,0,11230,3.8,0);  
Segment_Array[74].set_values(422110943,422111016,24.77,0,11230,0.1,0);  
Segment_Array[75].set_values(422111016,422111056,24.77,0,11230,2.2,0);  
Segment_Array[76].set_values(422111056,422111016,24.77,0,11230,2.3,0);  
Segment_Array[3075].set_values(422111016,422110943,24.77,0,11230,2.3,3);  
Segment_Array[3074].set_values(422110943,422110931,24.77,0,11230,2.2,3);  
Segment_Array[3073].set_values(422110931,422110931,24.77,0,11230,0.2,3);
```

```
Segment_Array[78].set_values(512113015,512113020,24.77,0,8450,0,0);  
Segment_Array[79].set_values(512113020,512113012,24.77,0,8450,1.6,0);  
Segment_Array[3002].set_values(512113012,512112975,17.98,0,8450,1.6,3);  
Segment_Array[3003].set_values(512112975,512112937,17.98,0,8450,2.1,3);  
Segment_Array[3004].set_values(512112937,522112930,17.98,0,8450,2,3);  
Segment_Array[3005].set_values(522112930,112112909,17.98,0,9315,0.2,3);  
Segment_Array[3006].set_values(112112909,112112918,17.98,0,9315,2,3);  
Segment_Array[3007].set_values(112112918,112112912,17.98,0,9315,1.3,3);  
Segment_Array[3008].set_values(112112912,112112910,17.98,0,9315,1.2,3);  
Segment_Array[3009].set_values(112112910,522112882,17.98,0,9315,0.3,3);  
Segment_Array[3010].set_values(522112882,242112829,17.98,0,9315,2.6,3);  
Segment_Array[3011].set_values(242112829,242112825,17.98,0,1650,3.2,3);  
Segment_Array[3012].set_values(242112825,242112595,17.98,0,1650,0.2,3);  
Segment_Array[3013].set_values(242112595,242112531,17.98,0,1650,12.1,3);  
Segment_Array[3014].set_values(242112531,242112520,17.98,0,1650,1.9,3);  
Segment_Array[3015].set_values(242112520,242112491,17.98,0,1650,1.3,3);  
Segment_Array[3016].set_values(242112491,242112482,17.98,0,980,1.1,3);  
Segment_Array[3017].set_values(242112482,242112476,17.98,0,980,0.2,3);  
Segment_Array[3018].set_values(242112476,242112345,17.98,0,980,0.3,3);  
Segment_Array[3019].set_values(242112345,242112335,17.98,0,980,7.3,3);  
Segment_Array[3020].set_values(242112335,242112280,17.98,0,980,1.3,3);  
Segment_Array[3021].set_values(242112280,242112279,17.98,0,9060,2.7,3);  
Segment_Array[3022].set_values(242112279,242112256,17.98,0,9060,0,3);  
Segment_Array[3023].set_values(242112256,242112231,17.98,0,9060,1,3);  
Segment_Array[3024].set_values(242112231,242112228,17.98,0,9060,0.7,3);  
Segment_Array[3025].set_values(242112228,242112216,17.98,0,9060,0,3);  
Segment_Array[3026].set_values(242112216,242112215,17.98,0,9060,0.8,3);  
Segment_Array[3027].set_values(242112215,242112213,17.98,0,9060,0.1,3);  
Segment_Array[3028].set_values(242112213,242112204,17.98,0,9060,0.1,3);
```

```

Segment_Array[3029].set_values(242112204,242112118,17.98,0,9060,0.3,3);
Segment_Array[3030].set_values(242112118,242112165,17.98,0,9060,4.7,3);
Segment_Array[3031].set_values(242112165,242112166,17.98,0,9060,2.5,3);
Segment_Array[3032].set_values(242112166,242112157,17.98,0,9060,0.1,3);
Segment_Array[3033].set_values(242112157,242112129,17.98,0,9060,0.5,3);
Segment_Array[3034].set_values(242112129,242111981,17.98,0,9060,1.3,3);
Segment_Array[3035].set_values(242111981,242111734,17.98,0,9060,11.2,3);
Segment_Array[3036].set_values(242111734,522111573,17.98,0,250,18.5,3);
Segment_Array[3037].set_values(522111573,102111464,17.98,0,250,18.4,3);
Segment_Array[3038].set_values(102111464,102111450,17.98,0,1170,10,3);
Segment_Array[3039].set_values(102111450,102111443,17.98,0,1170,2,3);
Segment_Array[3040].set_values(102111443,522111200,17.98,0,1170,0.2,3);
Segment_Array[3041].set_values(522111200,422111179,17.98,0,1170,9,3);
Segment_Array[3042].set_values(422111179,422110995,17.98,0,575,0.7,3);
Segment_Array[3043].set_values(422110995,422110962,17.98,0,575,12.7,3);
Segment_Array[3044].set_values(422110962,422110960,17.98,0,11230,1.9,3);
Segment_Array[3045].set_values(422110960,422110955,17.98,0,11230,0.2,3);
Segment_Array[3046].set_values(422110955,422110959,17.98,0,11230,0.5,3);
Segment_Array[3047].set_values(422110959,422110981,17.98,0,11230,0.2,3);
Segment_Array[3048].set_values(422110981,422111016,17.98,0,11230,0.7,3);
Segment_Array[3049].set_values(422111016,422110943,17.98,0,11230,1.3,3);
Segment_Array[3050].set_values(422110943,422110931,17.98,0,11230,2.3,3);
Segment_Array[3051].set_values(422110931,422110931,17.98,0,11230,0.1,3);

```

```

Route *Route_Array;
Route_Array = new Route[50];

```

```

int route0[73]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,22,
23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,
43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,
63,64,65,66,67,68,69,70,71,72};

```

```

int route1[74]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,
44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,1063,
1064,1065,1066,1067,1068,1069,1070,1071,1072,1073};

```

```

int route2[79]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,22,23,
24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,
44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
64,65,66,67,68,69,70,71,77,74,75,76,3075,3074,3073};

```

```

int route3[52]={78,79,3002,3003,3004,3005,3006,3007,3008,3009,3010,3011,3012,
3013,3014,3015,3016,3017,3018,3019,3020,3021,3022,3023,3024,
3025,3026,3027,3028,3029,3030,3031,3032,3033,3034,3035,3036,

```

```

3037,3038,3039,3040,3041,3042,3043,3044,3045,3046,3047,3048,
3049,3050,3051});

int numRoutes = 4; //number of routes to be analyzed
Route_Array[0].set_values(0, 73, route0);

Route_Array[1].set_values(1, 74, route1);

Route_Array[2].set_values(2, 79, route2);

Route_Array[3].set_values(3, 52, route3);

double hslArray[4][3] = {{0.0001, 0.001, 0.01}, //95% CI hsl red, low mean hi
                        {0.00001, 0.0001, 0.001}, //95% CI hsl orange
                        {0.000001, 0.00001, 0.0001}, //95% CI hsl yellow
                        {0.0000001, 0.000001, 0.00001} }; //95% CI hsl green

int hsl, cars;
double cost_life = 3700000;

double killed, p_hit, p_k_re, pop_exp;
double p_re_hit = 1.0/3; //probability of release given that a car is hit

double tons_CI_car = 90.0;
double grams_CI_per_ton = 907184.74;
double L50 = 210; //the dosage exposure of chlorine for 30 min for which
                //50% of the population dies
double sigma = 420 / pow(2*M_PI,0.5); //the standard deviation of the death
                //rate from chlorine exposure for 30 minutes
double percent_release = 0.50; //amount of chlorine release if an impacted
                //tank car ruptures
double radius = 800.0;

fp_out << "Route,#cars,HaulCost,Fatality,CostLife,#Injured," <<
        "CostInjury,VolReleased,CostCleanup," <<
        "maxFatality,maxCostLife,maxInjured,maxCostInjury," <<
        "maxVolReleased,maxCostCleanup,VarRiskCost," <<
        "avgRChslRperc05,avgRChslRperc50,avgRChslRperc95," <<
        "avgRChslOperc05,avgRChslOperc50,avgRChslOperc95," <<
        "avgRChslYperc05,avgRChslYperc50,avgRChslYperc95," <<
        "avgRChslGperc05,avgRChslGperc50,avgRChslGperc95," <<
        "maxUnweightedAccidentCost,maxRChslR05,maxRChslR50," <<
        "maxRChslR95," <<
        "maxRChslO05,maxRChslO50,maxRChslO95,maxRChslY05," <<
        "maxRChslY50,maxRChslY95,maxRChslG05,maxRChslG50," <<
        "maxRChslG95" << endl;

```

```

for (int i = 0; i < numRoutes; i++) //loop over routes
{
    double routeLength = 0.0;
    double total_pop_density = 0.0;
    for (int k = 0; k < Route_Array[i].getNumSegs(); k++)
    {
        routeLength += Segment_Array
            [Route_Array[i].getSegNum(k)].getDistMiles();
        total_pop_density += Segment_Array
            [Route_Array[i].getSegNum(k)].getPopSeg();
    }

    for (cars=1; cars<=10; cars ++)
    {
//      fp_out << "Route:," << i << ",#cars," << cars << endl << endl;
        double totalLivesLost = 0.0;
        double totalCostLife = 0.0;
        double totalInjured = 0.0;
        double totalCostInjured = 0.0;
        double totalVolReleased = 0.0;
        double totalCostCleanup = 0.0;
        double totalRiskCost = 0.0;
        double maxLivesLost = 0.0;
        double maxCostLife = 0.0;
        double maxInjured = 0.0;
        double maxCostInjured = 0.0;
        double maxVolReleased = 0.0;
        double maxCostCleanup = 0.0;
        double maxCost = 0.0;

        double totalHaulCost = 0.0;

        for (int k = 0; k < Route_Array[i].getNumSegs(); k++)
        {
            //begin calculations for route segment
            int segArrayNum = Route_Array[i].getSegNum(k);

            double haulcost = ( cars * Segment_Array
                [segArrayNum].getDistMiles() *
                Segment_Array
                [segArrayNum].getCost_CarMile() );

            //the following calculates the volume of the hemisphere of Cl2
            //around the release point site

```

```

double volume_hemisphere = 2*M_PI*pow(radius,3)/(3*0.001);

double Ldoes = p_re_hit * cars * tons_Cl_car * percent_release *
               grams_Cl_per_ton / 3.2 * 1000000 /
               volume_hemisphere;
double distMeters = Segment_Array[segArrayNum].getDistMeters();
if (distMeters <= 1600) distMeters = 1; //prevent divide by zero
   p_k_re = 1/(1+exp(-(Ldoes - L50)/sigma));
pop_exp = Segment_Array[segArrayNum].getPopSeg() * 320000 /2589988.11;
//1600m * 1600m = 320000 m^2
//1 square miles = 2 589 988.11 square meters (says google)
killed = p_k_re * pop_exp;
   double injured = pop_exp - killed;

double volume_released = p_re_hit * 90 * percent_release * cars;
   //above is tons Cl2 released

double cost_injured = log(volume_released) * 18000;
double cost_cleanup = 4000000 / 90.00 * volume_released;

//the following calculates the probability, given that there is an
//attack, that the attack will be on this segment
double p_Seg_Attack = Segment_Array[segArrayNum].getPopSeg() /
                    total_pop_density;

double seg_cost = (killed * cost_life + injured *
                  cost_injured + cost_cleanup);

totalLivesLost += p_Seg_Attack * killed;
totalCostLife += p_Seg_Attack * (killed * cost_life);
totalInjured += p_Seg_Attack * injured;
totalCostInjured += p_Seg_Attack * injured * cost_injured;
totalVolReleased += p_Seg_Attack * volume_released;
totalCostCleanup += p_Seg_Attack * cost_cleanup;
totalRiskCost += p_Seg_Attack * seg_cost;
totalHaulCost += haulcost;

if (maxCost < seg_cost) maxCost = seg_cost;

if (maxLivesLost < killed) maxLivesLost = killed;
if (maxCostLife < (killed * cost_life)) maxCostLife = killed *
                    cost_life;
if (maxInjured < injured) maxInjured = injured;
if (maxCostInjured < cost_injured) maxCostInjured = injured *
                    cost_injured;
if (maxVolReleased < volume_released) maxVolReleased =

```

```

                                volume_released;
    if (maxCostCleanup < cost_cleanup) maxCostCleanup = cost_cleanup;

} //end segment calculation

fp_out << i << "," << cars << "," << totalHaulCost << "," <<
    totalLivesLost << "," << totalCostLife << "," << totalInjured <<
    "," << totalCostInjured << "," << totalVolReleased << "," <<
    totalCostCleanup << "," << maxLivesLost << "," << maxCostLife <<
    "," << maxInjured << "," << maxCostInjured << "," <<
    maxVolReleased << "," << maxCostCleanup << "," << totalRiskCost
    << "," <<
    totalRiskCost*hslArray[0][0]+totalHaulCost << "," <<
    totalRiskCost*hslArray[0][1]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[0][2]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[1][0]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[1][1]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[1][2]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[2][0]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[2][1]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[2][2]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[3][0]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[3][1]+(totalHaulCost) << "," <<
    totalRiskCost*hslArray[3][2]+(totalHaulCost) << "," <<
    (maxCost + totalHaulCost) << "," << maxCost*hslArray[0][0]+
    (totalHaulCost) << "," << maxCost*hslArray[0][1]+(totalHaulCost)
    << "," << maxCost*hslArray[0][2]+(totalHaulCost) << "," <<
    maxCost*hslArray[1][0]+(totalHaulCost) << "," <<
    maxCost*hslArray[1][1]+(totalHaulCost) << "," <<
    maxCost*hslArray[1][2]+(totalHaulCost) << "," <<
    maxCost*hslArray[2][0]+(totalHaulCost) << "," <<
    maxCost*hslArray[2][1]+(totalHaulCost) << "," <<
    maxCost*hslArray[2][2]+(totalHaulCost) << "," <<
    maxCost*hslArray[3][0]+(totalHaulCost) << "," <<
    maxCost*hslArray[3][1]+(totalHaulCost) << "," <<
    maxCost*hslArray[3][2]+(totalHaulCost) << endl;
} //end cars loop
} //end route loop

cout << "-----program done-----"
    << endl << endl;
fp_out.close();
cin >> sillyVariableToKeepTheProgramWindowOpen;

} // end of main program

```

Appendix D: Stochastic Model

```
// The stochastic implementation of the risk cost model
// Please note that this implementation uses a uniform distribution in route
// segment selection. Further use should incorporate the proportion of
// of population density methodology used in the (unweighted) deterministic
// implementation.
// Authors: Matthew Albin, Ruben Luna, Danielle Martin, Danielle Obuchon, Andy
// Ramsay
// 20 April 2008
```

```
#include <iostream.h>
#include <math.h>
#include <fstream.h> // file I/O
#include <iomanip.h> // format manipulation
```

```
class Route {
    int routeNum; //unique route number
    int numSegs; //number of segments in the route--max is 50 (size of
                //array below)
    int segNums[100]; //segment numbers constituting route

public: void set_values (int in_routeNum, int in_numSegs, int in_segNums[])
    {
        routeNum = in_routeNum;
        numSegs = in_numSegs;
        for (int i = 0; i < numSegs; i++)
        {
            segNums[i] = in_segNums[i];
        }
    }
    void print_values (void)
    {
        cout << "Route " << routeNum << " contains " << numSegs <<
            " segments, segments ";
        int i, currentSeg;
        for (i = 0; i < numSegs-1; i++)
        {
            currentSeg = getSegNum(i);
            cout << currentSeg << ", ";
        }
        currentSeg = getSegNum(i);
        cout << currentSeg << "." << endl;
    }
    int getRouteNum (void) {return routeNum;}
    int getNumSegs (void) {return numSegs;}
}
```

```

    int getSegNum (int segNumCounter) {return segNums[segNumCounter];}
};

class Segment {
    int startNode;
    int endNode;
    double cost_car_mile;
    double PV;
    double pop_seg;
    double dist_miles;
    double dist_meters;
    int carrier;

public:
    void set_values (int in_startNode, int in_endNode, double in_cost_car_mile,
                    double in_PV, double in_pop_seg, double in_dist_miles,
                    int in_carrier)
    {
        startNode = in_startNode;
        endNode = in_endNode;
        cost_car_mile = in_cost_car_mile;
        PV = in_PV;
        pop_seg = in_pop_seg;
        dist_miles = in_dist_miles;
        dist_meters = dist_miles * 1609.344;
        carrier = in_carrier;
    }
    void print_values (void)
    {
        cout << " Start node is " << startNode << "." << endl;
        cout << " End node is " << endNode << "." << endl;
        cout << " Cost per car-mile is " << cost_car_mile << endl;
        cout << " # Points of vulnerability is " << PV << endl;
        cout << " Population along segment is " << pop_seg <<endl;
        cout << " Distance in miles is " << dist_miles << endl;
        cout << " Distance in meters is " << dist_meters << endl;
        cout << " Carrier is ";
        if (carrier == 0) cout << "Norfolk Southern";
        else if (carrier == 1) cout << "CPRS";
        else if (carrier == 2) cout << "CSX";
        else if (carrier == 3) cout << "CR";
        else cout << "unknown";
        cout << "." << endl;
    }
    double getCost_CarMile () {return cost_car_mile;}
    double getPV () {return PV;}
};

```

```
double getPopSeg () {return pop_seg;}
double getDistMiles () {return dist_miles;}
double getDistMeters () {return dist_meters;}
};

void main()
{
int sillyVariableToKeepTheProgramWindowOpen;
ofstream fp_out;
fp_out.open("stoversion3_trial10.txt", ios::out);
fp_out << setprecision(14);

Segment *Segment_Array;
Segment_Array = new Segment[10000];

Segment_Array[0].set_values(512113015,512113023,24.77,0,8450,0,0);
Segment_Array[1].set_values(512113023,512113027,24.77,0,2450,1.7,0);;
Segment_Array[2].set_values(512113027,512113038,24.77,0,2450,2.4,0);
Segment_Array[3].set_values(512113038,512113037,24.77,0,2450,13.8,0);
Segment_Array[4].set_values(512113037,512113054,24.77,0,3530,0.5,0);
Segment_Array[5].set_values(512113054,512113058,24.77,0,3530,1.4,0);
Segment_Array[6].set_values(512113058,512113075,24.77,0,4130,0.2,0);
Segment_Array[7].set_values(512113075,512113073,24.77,0,4130,2.1,0);
Segment_Array[8].set_values(512113073,512113062,24.77,0,4130,0.3,0);
Segment_Array[9].set_values(512113062,512113000,24.77,0,85,1.1,0);
Segment_Array[10].set_values(512113000,512112885,24.77,0,85,12.7,0);
Segment_Array[11].set_values(512112885,512112850,24.77,0,85,28.2,0);
Segment_Array[12].set_values(512112850,512112848,24.77,0,85,7.2,0);
Segment_Array[13].set_values(512112848,512112839,24.77,0,85,0.9,0);
Segment_Array[15].set_values(512112836,512112746,24.77,0,140,0.1,0);
Segment_Array[16].set_values(512112746,522112389,24.77,0,140,7,0);
Segment_Array[17].set_values(522112389,542112172,24.77,0,240,16.6,0);
Segment_Array[18].set_values(542112172,542112069,24.77,0,240,8,0);
Segment_Array[19].set_values(542112069,522111970,24.77,0,290,4.1,0);
Segment_Array[20].set_values(522111970,242111643,24.77,0,290,6.6,0);
Segment_Array[21].set_values(242111643,242111635,24.77,0,290,15.3,0);
Segment_Array[22].set_values(242111635,522111481,24.77,0,290,0.4,0);
Segment_Array[23].set_values(522111481,422110664,24.77,0,290,6.5,0);
Segment_Array[24].set_values(422110664,422110660,24.77,0,400,26.3,0);
Segment_Array[25].set_values(22110660,422110649,24.77,0,400,0.1,0);
Segment_Array[26].set_values(422110649,422110515,24.77,0,400,0.6,0);
Segment_Array[27].set_values(422110515,422110471,24.77,0,400,19,0);
Segment_Array[28].set_values(422110471,422110454,24.77,0,400,8.7,0);
Segment_Array[29].set_values(422110454,422110360,24.77,0,400,2.2,0);
Segment_Array[30].set_values(422110360,422110296,24.77,0,400,4.4,0);
```

```
Segment_Array[31].set_values(422110296,422110267,24.77,0,400,3,0);
Segment_Array[32].set_values(422110267,422110256,24.77,0,400,1.8,0);
Segment_Array[33].set_values(422110256,422110254,24.77,0,480,0.8,0);
Segment_Array[34].set_values(422110254,422110258,24.77,0,480,0.2,0);
Segment_Array[35].set_values(422110258,422110257,24.77,0,480,0.3,0);
Segment_Array[36].set_values(422110257,422110227,24.77,0,480,0.5,0);
Segment_Array[37].set_values(422110227,422110228,24.77,0,480,7,0);
Segment_Array[38].set_values(422110228,422110230,24.77,0,480,0.6,0);
Segment_Array[39].set_values(422110230,422110150,24.77,0,480,0.2,0);
Segment_Array[40].set_values(422110150,422110059,24.77,0,335,6.3,0);
Segment_Array[41].set_values(422110059,422110058,24.77,0,335,9.5,0);
Segment_Array[42].set_values(422110058,422110029,24.77,0,335,0.2,0);
Segment_Array[43].set_values(422110029,422110100,24.77,0,435,12.9,0);
Segment_Array[44].set_values(422110100,422110072,24.77,0,435,9.5,0);
Segment_Array[45].set_values(422110072,422110068,24.77,0,435,3.6,0);
Segment_Array[46].set_values(422110068,422110066,24.77,0,435,0.5,0);
Segment_Array[47].set_values(422110066,422110153,24.77,0,435,0.2,0);
Segment_Array[48].set_values(422110153,422110124,24.77,0,435,3.5,0);
Segment_Array[49].set_values(422110124,422110219,24.77,0,435,0.9,0);
Segment_Array[50].set_values(422110219,422110221,24.77,0,435,6.2,0);
Segment_Array[51].set_values(422110221,422110270,24.77,0,435,0.2,0);
Segment_Array[52].set_values(422110270,422110271,24.77,0,552,6.7,0);
Segment_Array[53].set_values(422110271,422110331,24.77,0,552,2,0);
Segment_Array[54].set_values(422110331,422110457,24.77,0,552,5,0);
Segment_Array[55].set_values(422110457,422110460,24.77,0,552,7,0);
Segment_Array[56].set_values(422110460,422110485,24.77,0,552,0,0);
Segment_Array[57].set_values(422110485,422110522,24.77,0,552,1,0);
Segment_Array[58].set_values(422110522,422110528,24.77,0,552,2.5,0);
Segment_Array[59].set_values(422110528,422110558,24.77,0,552,0.3,0);
Segment_Array[60].set_values(422110558,422110543,24.77,0,552,1.3,0);
Segment_Array[61].set_values(422110543,422110532,24.77,0,552,3.1,0);
Segment_Array[62].set_values(422110532,422110540,24.77,0,552,1.4,0);
Segment_Array[63].set_values(422110540,422110552,24.77,0,552,1.3,0);
Segment_Array[64].set_values(422110552,422110566,24.77,0,552,1.1,0);
Segment_Array[65].set_values(422110566,422110604,24.77,0,552,1,0);
Segment_Array[66].set_values(422110604,422110673,24.77,0,552,1.4,0);
Segment_Array[67].set_values(422110673,422110690,24.77,0,552,5.6,0);
Segment_Array[68].set_values(422110690,422110733,24.77,0,552,0.5,0);
Segment_Array[69].set_values(422110733,422110744,24.77,0,552,2.1,0);
Segment_Array[70].set_values(422110744,422110754,24.77,0,552,0.7,0);
Segment_Array[71].set_values(422110754,422110813,24.77,0,11230,0.2,0);
Segment_Array[72].set_values(422110813,422110931,24.77,0,11230,1.3,0);
Segment_Array[73].set_values(422110931,422110931,24.77,0,11230,3.8,0);
```

```
Segment_Array[1063].set_values(422110540,422110552,21.37,0,552,1.3,1);
```

```
Segment_Array[1064].set_values(422110552,422110566,21.37,0,552,1.1,1);  
Segment_Array[1065].set_values(422110566,422110604,21.37,0,552,1,1);  
Segment_Array[1066].set_values(422110604,422110673,21.37,0,552,1.4,1);  
Segment_Array[1067].set_values(422110673,422110690,21.37,0,552,5.6,1);  
Segment_Array[1068].set_values(422110690,422110733,21.37,0,552,0.5,1);  
Segment_Array[1069].set_values(422110733,422110744,21.37,0,552,2.1,1);  
Segment_Array[1070].set_values(422110744,422110754,21.37,0,552,0.7,1);  
Segment_Array[1071].set_values(422110754,422110813,21.37,0,11230,0.2,1);  
Segment_Array[1072].set_values(422110813,422110931,21.37,0,11230,1.3,1);  
Segment_Array[1073].set_values(422110931,422110931,21.37,0,11230,3.8,1);
```

```
Segment_Array[77].set_values(422110931,422110943,24.77,0,11230,3.8,0);  
Segment_Array[74].set_values(422110943,422111016,24.77,0,11230,0.1,0);  
Segment_Array[75].set_values(422111016,422111056,24.77,0,11230,2.2,0);  
Segment_Array[76].set_values(422111056,422111016,24.77,0,11230,2.3,0);  
Segment_Array[3075].set_values(422111016,422110943,24.77,0,11230,2.3,3);  
Segment_Array[3074].set_values(422110943,422110931,24.77,0,11230,2.2,3);  
Segment_Array[3073].set_values(422110931,422110931,24.77,0,11230,0.2,3);
```

```
Segment_Array[78].set_values(512113015,512113020,24.77,0,8450,0,0);  
Segment_Array[79].set_values(512113020,512113012,24.77,0,8450,1.6,0);  
Segment_Array[3002].set_values(512113012,512112975,17.98,0,8450,1.6,3);  
Segment_Array[3003].set_values(512112975,512112937,17.98,0,8450,2.1,3);  
Segment_Array[3004].set_values(512112937,522112930,17.98,0,8450,2,3);  
Segment_Array[3005].set_values(522112930,112112909,17.98,0,9315,0.2,3);  
Segment_Array[3006].set_values(112112909,112112918,17.98,0,9315,2,3);  
Segment_Array[3007].set_values(112112918,112112912,17.98,0,9315,1.3,3);  
Segment_Array[3008].set_values(112112912,112112910,17.98,0,9315,1.2,3);  
Segment_Array[3009].set_values(112112910,522112882,17.98,0,9315,0.3,3);  
Segment_Array[3010].set_values(522112882,242112829,17.98,0,9315,2.6,3);  
Segment_Array[3011].set_values(242112829,242112825,17.98,0,1650,3.2,3);  
Segment_Array[3012].set_values(242112825,242112595,17.98,0,1650,0.2,3);  
Segment_Array[3013].set_values(242112595,242112531,17.98,0,1650,12.1,3);  
Segment_Array[3014].set_values(242112531,242112520,17.98,0,1650,1.9,3);  
Segment_Array[3015].set_values(242112520,242112491,17.98,0,1650,1.3,3);  
Segment_Array[3016].set_values(242112491,242112482,17.98,0,980,1.1,3);  
Segment_Array[3017].set_values(242112482,242112476,17.98,0,980,0.2,3);  
Segment_Array[3018].set_values(242112476,242112345,17.98,0,980,0.3,3);  
Segment_Array[3019].set_values(242112345,242112335,17.98,0,980,7.3,3);  
Segment_Array[3020].set_values(242112335,242112280,17.98,0,980,1.3,3);  
Segment_Array[3021].set_values(242112280,242112279,17.98,0,9060,2.7,3);  
Segment_Array[3022].set_values(242112279,242112256,17.98,0,9060,0,3);  
Segment_Array[3023].set_values(242112256,242112231,17.98,0,9060,1,3);  
Segment_Array[3024].set_values(242112231,242112228,17.98,0,9060,0.7,3);
```

```

Segment_Array[3025].set_values(242112228,242112216,17.98,0,9060,0,3);
Segment_Array[3026].set_values(242112216,242112215,17.98,0,9060,0,8,3);
Segment_Array[3027].set_values(242112215,242112213,17.98,0,9060,0,1,3);
Segment_Array[3028].set_values(242112213,242112204,17.98,0,9060,0,1,3);
Segment_Array[3029].set_values(242112204,242112118,17.98,0,9060,0,3,3);
Segment_Array[3030].set_values(242112118,242112165,17.98,0,9060,4,7,3);
Segment_Array[3031].set_values(242112165,242112166,17.98,0,9060,2,5,3);
Segment_Array[3032].set_values(242112166,242112157,17.98,0,9060,0,1,3);
Segment_Array[3033].set_values(242112157,242112129,17.98,0,9060,0,5,3);
Segment_Array[3034].set_values(242112129,242111981,17.98,0,9060,1,3,3);
Segment_Array[3035].set_values(242111981,242111734,17.98,0,9060,11,2,3);
Segment_Array[3036].set_values(242111734,522111573,17.98,0,250,18,5,3);
Segment_Array[3037].set_values(522111573,102111464,17.98,0,250,18,4,3);
Segment_Array[3038].set_values(102111464,102111450,17.98,0,1170,10,3);
Segment_Array[3039].set_values(102111450,102111443,17.98,0,1170,2,3);
Segment_Array[3040].set_values(102111443,522111200,17.98,0,1170,0,2,3);
Segment_Array[3041].set_values(522111200,422111179,17.98,0,1170,9,3);
Segment_Array[3042].set_values(422111179,422110995,17.98,0,575,0,7,3);
Segment_Array[3043].set_values(422110995,422110962,17.98,0,575,12,7,3);
Segment_Array[3044].set_values(422110962,422110960,17.98,0,11230,1,9,3);
Segment_Array[3045].set_values(422110960,422110955,17.98,0,11230,0,2,3);
Segment_Array[3046].set_values(422110955,422110959,17.98,0,11230,0,5,3);
Segment_Array[3047].set_values(422110959,422110981,17.98,0,11230,0,2,3);
Segment_Array[3048].set_values(422110981,422111016,17.98,0,11230,0,7,3);
Segment_Array[3049].set_values(422111016,422110943,17.98,0,11230,1,3,3);
Segment_Array[3050].set_values(422110943,422110931,17.98,0,11230,2,3,3);
Segment_Array[3051].set_values(422110931,422110931,17.98,0,11230,0,1,3);

```

```

Route *Route_Array;
Route_Array = new Route[50];

```

```

int route0[73]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,22,
23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,
43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,
63,64,65,66,67,68,69,70,71,72};

```

```

int route1[74]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,
44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,1063,
1064,1065,1066,1067,1068,1069,1070,1071,1072,1073};

```

```

int route2[79]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,22,23,
24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,
44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
64,65,66,67,68,69,70,71,77,74,75,76,3075,3074,3073};

```

```
int route3[52]={78,79,3002,3003,3004,3005,3006,3007,3008,3009,3010,3011,3012,
  3013,3014,3015,3016,3017,3018,3019,3020,3021,3022,3023,3024,
  3025,3026,3027,3028,3029,3030,3031,3032,3033,3034,3035,3036,
  3037,3038,3039,3040,3041,3042,3043,3044,3045,3046,3047,3048,
  3049,3050,3051};
```

```
int numRoutes = 4; //number of routes to be analyzed
Route_Array[0].set_values(0, 73, route0);
```

```
Route_Array[1].set_values(1, 74, route1);
```

```
Route_Array[2].set_values(2, 79, route2);
```

```
Route_Array[3].set_values(3, 52, route3);
```

```
double hslArray[4][3] = {{0.0001, 0.001, 0.01}, //95% CI hsl red, low mean hi
  {0.00001, 0.0001, 0.001}, //95% CI hsl orange
  {0.000001, 0.00001, 0.0001}, //95% CI hsl yellow
  {0.0000001, 0.000001, 0.00001} }; //95% CI hsl green
```

```
int hsl, cars;
double cost_life = 3700000;
```

```
double killed, p_hit, p_k_re, pop_exp;
double p_re_hit = 1.0/3; //probability of release given that a car is hit
```

```
double tons_Cl_car = 90.0;
double grams_Cl_per_ton = 907184.74;
double L50 = 210; //the dosage exposure of chlorine for 30 min for which
  //50% of the population dies
double sigma = 420 / pow(2*M_PI,0.5); //the standard deviation of the death
  //rate from chlorine exposure for 30 minutes
double percent_release = 0.50; //amount of chlorine release if an impacted
  //tank car ruptures
double radius = 800.0;
```

```
fp_out << "Route,#cars,HaulCost,avgFatality,avgCostLife," <<
  "avg#Injured,avgCostInjury,avgVolReleased," <<
  "avgCostCleanup," <<
  "maxFatality,maxCostLife,maxInjured,maxCostInjury," <<
  "maxVolReleased,maxCostCleanup,avgVarRiskCost," <<
  "avgRChslRperc05,avgRChslRperc50,avgRChslRperc95," <<
  "avgRChslOperc05,avgRChslOperc50,avgRChslOperc95," <<
  "avgRChslYperc05,avgRChslYperc50,avgRChslYperc95," <<
  "avgRChslGperc05,avgRChslGperc50,avgRChslGperc95," <<
```

```

    "maxUnweightedAccidentCost,maxRChslR05,maxRChslR50," <<
    "maxRChslR95,maxRChslO05,maxRChslO50,maxRChslO95,"
    <<"maxRChslY05,maxRChslY50,maxRChslY95,maxRChslG05," <<
    "maxRChslG50,maxRChslG95" << endl;

```

```

for (int i = 0; i < numRoutes; i++) //loop over routes
{
    double routeLength = 0.0;
    for (int k = 0; k < Route_Array[i].getNumSegs(); k++)
    {
        routeLength += Segment_Array
            [Route_Array[i].getSegNum(k)].getDistMiles();
    }

    for (cars=1; cars<=10; cars ++)
    {
        double totalLivesLost = 0.0;
        double totalCostLife = 0.0;
        double totalInjured = 0.0;
        double totalCostInjured = 0.0;
        double totalVolReleased = 0.0;
        double totalCostCleanup = 0.0;
        double totalRiskCost = 0.0;
        double maxLivesLost = 0.0;
        double maxCostLife = 0.0;
        double maxInjured = 0.0;
        double maxCostInjured = 0.0;
        double maxVolReleased = 0.0;
        double maxCostCleanup = 0.0;
        double maxRiskCost = 0.0;

        double haulcost = 0.0;
        for (int k = 0; k < Route_Array[i].getNumSegs(); k++)
        {
            haulcost += ( cars * Segment_Array
                [Route_Array[i].getSegNum(k)].getDistMiles() *
                Segment_Array
                [Route_Array[i].getSegNum(k)].getCost_CarMile() );
        }

        int randSeg;
        int numCycles = 10000000;
        //Use the below line to move to a different set of random numbers
        //depending on which trial of numCycle runs you are currently doing.
        for (int z = 0; z < (9*numCycles); z++) int throwaway = rand();
        for (int cycle = 1; cycle <= numCycles; cycle++)

```

```

{
//begin calculations for route segment
randSeg = int (floor ( ((0.0 + rand())/RAND_MAX) *
    (Route_Array[i].getNumSegs() ) );
if (randSeg == (Route_Array[i].getNumSegs()))
{
    randSeg = Route_Array[i].getNumSegs()-1;
}
int segArrayNum = Route_Array[i].getSegNum(randSeg);

double volume_hemisphere = 2*M_PI*pow(radius,3)/(3*0.001);
//the above is the volume of the hemisphere of Cl2
//around the release point site
double Ldoes = p_re_hit * cars * tons_Cl_car * percent_release *
    grams_Cl_per_ton / 3.2 * 1000000 /
    volume_hemisphere;
double distMeters = Segment_Array[segArrayNum].getDistMeters();
if (distMeters <= 1600) distMeters = 1; //prevent divide by zero
    p_k_re = 1/(1+exp(-(Ldoes - L50)/sigma));
pop_exp = Segment_Array[segArrayNum].getPopSeg() * 320000 /2589988.11;
//1600m * 1600m = 320000 m^2
//1 square miles = 2 589 988.11 square meters (says google)
killed = p_k_re * pop_exp;
    double injured = pop_exp - killed;

double volume_released = p_re_hit * 90 * percent_release * cars;
//above is the tons Cl2 released

double cost_injured = log(volume_released) * 18000;
double cost_cleanup = 4000000 / 90.00 * volume_released;
double seg_cost = killed * cost_life + injured * cost_injured +
    cost_cleanup;
totalLivesLost += killed;
totalCostLife += (killed * cost_life);
totalInjured += injured;
totalCostInjured += injured * cost_injured;
totalVolReleased += volume_released;
totalCostCleanup += cost_cleanup;
totalRiskCost += seg_cost;
if (maxLivesLost < killed) maxLivesLost = killed;
if (maxCostLife < (killed * cost_life)) maxCostLife = killed * cost_life;
if (maxInjured < injured) maxInjured = injured;
if (maxCostInjured < cost_injured) maxCostInjured = (injured*cost_injured);
if (maxVolReleased < volume_released) maxVolReleased = volume_released;
if (maxCostCleanup < cost_cleanup) maxCostCleanup = cost_cleanup;
if (maxRiskCost < seg_cost) maxRiskCost = seg_cost;

```

```

} //end cycle loop

fp_out << i << ", " << cars << ", " << haulcost << ", " <<
    totalLivesLost/numCycles << ", " << totalCostLife/numCycles <<
    ", " << totalInjured/numCycles << ", " <<
    totalCostInjured/numCycles << ", " << totalVolReleased/numCycles
    << ", " << totalCostCleanup/numCycles << ", " << maxLivesLost <<
    ", " << maxCostLife << ", " << maxInjured << ", " << maxCostInjured
    << ", " << maxVolReleased << ", " << maxCostCleanup << ", " <<
    totalRiskCost/numCycles << ", " <<
    totalRiskCost/numCycles*hslArray[0][0]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[0][1]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[0][2]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[1][0]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[1][1]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[1][2]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[2][0]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[2][1]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[2][2]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[3][0]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[3][1]+(haulcost) << ", " <<
    totalRiskCost/numCycles*hslArray[3][2]+(haulcost) << ", " <<
    (maxRiskCost + haulcost) << ", " <<
    maxRiskCost*hslArray[0][0]+(haulcost) << ", " <<
    maxRiskCost*hslArray[0][1]+(haulcost) << ", " <<
    maxRiskCost*hslArray[0][2]+(haulcost) << ", " <<
    maxRiskCost*hslArray[1][0]+(haulcost) << ", " <<
    maxRiskCost*hslArray[1][1]+(haulcost) << ", " <<
    maxRiskCost*hslArray[1][2]+(haulcost) << ", " <<
    maxRiskCost*hslArray[2][0]+(haulcost) << ", " <<
    maxRiskCost*hslArray[2][1]+(haulcost) << ", " <<
    maxRiskCost*hslArray[2][2]+(haulcost) << ", " <<
    maxRiskCost*hslArray[3][0]+(haulcost) << ", " <<
    maxRiskCost*hslArray[3][1]+(haulcost) << ", " <<
    maxRiskCost*hslArray[3][2]+(haulcost) << endl;
} //end cars loop
} //end route loop

cout << "-----Program done-----"
    << endl << endl;
fp_out.close();
cin >> sillyVariableToKeepTheProgramWindowOpen;

} // end of main program

```

Appendix E: Weighted Deterministic Model

```
// The following deterministic program implements a weighting scheme within the
// proposed model. An earlier version of the model, it assumes that an attack
// has occurred. This method of incorporating weights was discarded because
// weighting is implicit in the cost functions of physical transport, death,
// injury, and remediation.
// Authors: Matthew Albin, Ruben Luna, Danielle Martin, Danielle Obuchon, Andy
// Ramsay
// 7 April 2008
```

```
#include <iostream.h>
#include <math.h>
#include <fstream.h> // file I/O
#include <iomanip.h> // format manipulation
```

```
class Route {
    int routeNum; //unique route number
    int numSegs; //number of segments in the route--max is 50 (size of
                //array below)
    int segNums[100]; //segment numbers constituting route

public: void set_values (int in_routeNum, int in_numSegs, int in_segNums[])
    {
        routeNum = in_routeNum;
        numSegs = in_numSegs;
        for (int i = 0; i < numSegs; i++)
        {
            segNums[i] = in_segNums[i];
        }
    }
    void print_values (void)
    {
        cout << "Route " << routeNum << " contains " << numSegs <<
            " segments, segments ";
        int i, currentSeg;
        for (i = 0; i < numSegs-1; i++)
        {
            currentSeg = getSegNum(i);
            cout << currentSeg << ", ";
        }
        currentSeg = getSegNum(i);
        cout << currentSeg << "." << endl;
    }
    int getRouteNum (void) {return routeNum;}
};
```

```

    int getNumSegs (void) {return numSegs;}
    int getSegNum (int segNumCounter) {return segNums[segNumCounter];}
};

class Segment {
    int startNode;
    int endNode;
    double cost_car_mile;
    double PV;
    double pop_seg;
    double dist_miles;
    double dist_meters;
    int carrier;

public:
    void set_values (int in_startNode, int in_endNode, double in_cost_car_mile,
                    double in_PV, double in_pop_seg, double in_dist_miles,
                    int in_carrier)
    {
        startNode = in_startNode;
        endNode = in_endNode;
        cost_car_mile = in_cost_car_mile;
        PV = in_PV;
        pop_seg = in_pop_seg;
        dist_miles = in_dist_miles;
        dist_meters = dist_miles * 1609.344;
        carrier = in_carrier;
    }
    void print_values (void)
    {
        cout << " Start node is " << startNode << "." << endl;
        cout << " End node is " << endNode << "." << endl;
        cout << " Cost per car-mile is " << cost_car_mile << endl;
        cout << " # Points of vulnerability is " << PV << endl;
        cout << " Population along segment is " << pop_seg << endl;
        cout << " Distance in miles is " << dist_miles << endl;
        cout << " Distance in meters is " << dist_meters << endl;
        cout << " Carrier is ";
        if (carrier == 0) cout << "Norfolk Southern";
        else if (carrier == 1) cout << "CPRS";
        else if (carrier == 2) cout << "CSX";
        else if (carrier == 3) cout << "CR";
        else cout << "unknown";
        cout << "." << endl;
    }
    double getCost_CarMile () {return cost_car_mile;}

```

```

    double getPV () {return PV;}
    double getPopSeg () {return pop_seg;}
    double getDistMiles () {return dist_miles;}
    double getDistMeters () {return dist_meters;}
};

void main()
{
    int sillyVariableToKeepTheProgramWindowOpen;
    ofstream fp_out;
    fp_out.open("wwwRRoutput_RR_Weights.txt", ios::out);
    fp_out << setprecision(14);

    int numWeights = 4;

    double *weightArray = new double[5];
    double *originalWeightArray = new double[5];
    originalWeightArray[0] = .30; //haul cost weight
    originalWeightArray[1] = .20; //death weight
    originalWeightArray[2] = .20; //injury weight
    originalWeightArray[3] = .30; //cleanup weight

    for (int i = 0; i < numWeights; i++) weightArray[i] = originalWeightArray[i];

    Segment *Segment_Array;
    Segment_Array = new Segment[10000];

    Segment_Array[0].set_values(512113015,512113023,24.77,0,8450,0,0);
    Segment_Array[1].set_values(512113023,512113027,24.77,0,2450,1.7,0);
    Segment_Array[2].set_values(512113027,512113038,24.77,0,2450,2.4,0);
    Segment_Array[3].set_values(512113038,512113037,24.77,0,2450,13.8,0);
    Segment_Array[4].set_values(512113037,512113054,24.77,0,3530,0.5,0);
    Segment_Array[5].set_values(512113054,512113058,24.77,0,3530,1.4,0);
    Segment_Array[6].set_values(512113058,512113075,24.77,0,4130,0.2,0);
    Segment_Array[7].set_values(512113075,512113073,24.77,0,4130,2.1,0);
    Segment_Array[8].set_values(512113073,512113062,24.77,0,4130,0.3,0);
    Segment_Array[9].set_values(512113062,512113000,24.77,0,85,1.1,0);
    Segment_Array[10].set_values(512113000,512112885,24.77,0,85,12.7,0);
    Segment_Array[11].set_values(512112885,512112850,24.77,0,85,28.2,0);
    Segment_Array[12].set_values(512112850,512112848,24.77,0,85,7.2,0);
    Segment_Array[13].set_values(512112848,512112839,24.77,0,85,0.9,0);
    Segment_Array[15].set_values(512112836,512112746,24.77,0,140,0.1,0);
    Segment_Array[16].set_values(512112746,522112389,24.77,0,140,7,0);
    Segment_Array[17].set_values(522112389,542112172,24.77,0,240,16.6,0);
    Segment_Array[18].set_values(542112172,542112069,24.77,0,240,8,0);

```

```
Segment_Array[19].set_values(542112069,522111970,24.77,0,290,4.1,0);
Segment_Array[20].set_values(522111970,242111643,24.77,0,290,6.6,0);
Segment_Array[21].set_values(242111643,242111635,24.77,0,290,15.3,0);
Segment_Array[22].set_values(242111635,522111481,24.77,0,290,0.4,0);
Segment_Array[23].set_values(522111481,422110664,24.77,0,290,6.5,0);
Segment_Array[24].set_values(422110664,422110660,24.77,0,400,26.3,0);
Segment_Array[25].set_values(22110660,422110649,24.77,0,400,0.1,0);
Segment_Array[26].set_values(422110649,422110515,24.77,0,400,0.6,0);
Segment_Array[27].set_values(422110515,422110471,24.77,0,400,19,0);
Segment_Array[28].set_values(422110471,422110454,24.77,0,400,8.7,0);
Segment_Array[29].set_values(422110454,422110360,24.77,0,400,2.2,0);
Segment_Array[30].set_values(422110360,422110296,24.77,0,400,4.4,0);
Segment_Array[31].set_values(422110296,422110267,24.77,0,400,3,0);
Segment_Array[32].set_values(422110267,422110256,24.77,0,400,1.8,0);
Segment_Array[33].set_values(422110256,422110254,24.77,0,480,0.8,0);
Segment_Array[34].set_values(422110254,422110258,24.77,0,480,0.2,0);
Segment_Array[35].set_values(422110258,422110257,24.77,0,480,0.3,0);
Segment_Array[36].set_values(422110257,422110227,24.77,0,480,0.5,0);
Segment_Array[37].set_values(422110227,422110228,24.77,0,480,7,0);
Segment_Array[38].set_values(422110228,422110230,24.77,0,480,0.6,0);
Segment_Array[39].set_values(422110230,422110150,24.77,0,480,0.2,0);
Segment_Array[40].set_values(422110150,422110059,24.77,0,335,6.3,0);
Segment_Array[41].set_values(422110059,422110058,24.77,0,335,9.5,0);
Segment_Array[42].set_values(422110058,422110029,24.77,0,335,0.2,0);
Segment_Array[43].set_values(422110029,422110100,24.77,0,435,12.9,0);
Segment_Array[44].set_values(422110100,422110072,24.77,0,435,9.5,0);
Segment_Array[45].set_values(422110072,422110068,24.77,0,435,3.6,0);
Segment_Array[46].set_values(422110068,422110066,24.77,0,435,0.5,0);
Segment_Array[47].set_values(422110066,422110153,24.77,0,435,0.2,0);
Segment_Array[48].set_values(422110153,422110124,24.77,0,435,3.5,0);
Segment_Array[49].set_values(422110124,422110219,24.77,0,435,0.9,0);
Segment_Array[50].set_values(422110219,422110221,24.77,0,435,6.2,0);
Segment_Array[51].set_values(422110221,422110270,24.77,0,435,0.2,0);
Segment_Array[52].set_values(422110270,422110271,24.77,0,552,6.7,0);
Segment_Array[53].set_values(422110271,422110331,24.77,0,552,2,0);
Segment_Array[54].set_values(422110331,422110457,24.77,0,552,5,0);
Segment_Array[55].set_values(422110457,422110460,24.77,0,552,7,0);
Segment_Array[56].set_values(422110460,422110485,24.77,0,552,0,0);
Segment_Array[57].set_values(422110485,422110522,24.77,0,552,1,0);
Segment_Array[58].set_values(422110522,422110528,24.77,0,552,2.5,0);
Segment_Array[59].set_values(422110528,422110558,24.77,0,552,0.3,0);
Segment_Array[60].set_values(422110558,422110543,24.77,0,552,1.3,0);
Segment_Array[61].set_values(422110543,422110532,24.77,0,552,3.1,0);
Segment_Array[62].set_values(422110532,422110540,24.77,0,552,1.4,0);
Segment_Array[63].set_values(422110540,422110552,24.77,0,552,1.3,0);
Segment_Array[64].set_values(422110552,422110566,24.77,0,552,1.1,0);
```

```
Segment_Array[65].set_values(422110566,422110604,24.77,0,552,1,0);  
Segment_Array[66].set_values(422110604,422110673,24.77,0,552,1.4,0);  
Segment_Array[67].set_values(422110673,422110690,24.77,0,552,5.6,0);  
Segment_Array[68].set_values(422110690,422110733,24.77,0,552,0.5,0);  
Segment_Array[69].set_values(422110733,422110744,24.77,0,552,2.1,0);  
Segment_Array[70].set_values(422110744,422110754,24.77,0,552,0.7,0);  
Segment_Array[71].set_values(422110754,422110813,24.77,0,11230,0.2,0);  
Segment_Array[72].set_values(422110813,422110931,24.77,0,11230,1.3,0);  
Segment_Array[73].set_values(422110931,422110931,24.77,0,11230,3.8,0);
```

```
Segment_Array[1063].set_values(422110540,422110552,21.37,0,552,1.3,1);  
Segment_Array[1064].set_values(422110552,422110566,21.37,0,552,1.1,1);  
Segment_Array[1065].set_values(422110566,422110604,21.37,0,552,1,1);  
Segment_Array[1066].set_values(422110604,422110673,21.37,0,552,1.4,1);  
Segment_Array[1067].set_values(422110673,422110690,21.37,0,552,5.6,1);  
Segment_Array[1068].set_values(422110690,422110733,21.37,0,552,0.5,1);  
Segment_Array[1069].set_values(422110733,422110744,21.37,0,552,2.1,1);  
Segment_Array[1070].set_values(422110744,422110754,21.37,0,552,0.7,1);  
Segment_Array[1071].set_values(422110754,422110813,21.37,0,11230,0.2,1);  
Segment_Array[1072].set_values(422110813,422110931,21.37,0,11230,1.3,1);  
Segment_Array[1073].set_values(422110931,422110931,21.37,0,11230,3.8,1);
```

```
Segment_Array[77].set_values(422110931,422110943,24.77,0,11230,3.8,0);  
Segment_Array[74].set_values(422110943,422111016,24.77,0,11230,0.1,0);  
Segment_Array[75].set_values(422111016,422111056,24.77,0,11230,2.2,0);  
Segment_Array[76].set_values(422111056,422111016,24.77,0,11230,2.3,0);  
Segment_Array[3075].set_values(422111016,422110943,24.77,0,11230,2.3,3);  
Segment_Array[3074].set_values(422110943,422110931,24.77,0,11230,2.2,3);  
Segment_Array[3073].set_values(422110931,422110931,24.77,0,11230,0.2,3);
```

```
Segment_Array[78].set_values(512113015,512113020,24.77,0,8450,0,0);  
Segment_Array[79].set_values(512113020,512113012,24.77,0,8450,1.6,0);  
Segment_Array[3002].set_values(512113012,512112975,17.98,0,8450,1.6,3);  
Segment_Array[3003].set_values(512112975,512112937,17.98,0,8450,2.1,3);  
Segment_Array[3004].set_values(512112937,522112930,17.98,0,8450,2,3);  
Segment_Array[3005].set_values(522112930,112112909,17.98,0,9315,0.2,3);  
Segment_Array[3006].set_values(112112909,112112918,17.98,0,9315,2,3);  
Segment_Array[3007].set_values(112112918,112112912,17.98,0,9315,1.3,3);  
Segment_Array[3008].set_values(112112912,112112910,17.98,0,9315,1.2,3);  
Segment_Array[3009].set_values(112112910,522112882,17.98,0,9315,0.3,3);  
Segment_Array[3010].set_values(522112882,242112829,17.98,0,9315,2.6,3);  
Segment_Array[3011].set_values(242112829,242112825,17.98,0,1650,3.2,3);  
Segment_Array[3012].set_values(242112825,242112595,17.98,0,1650,0.2,3);
```

```
Segment_Array[3013].set_values(242112595,242112531,17.98,0,1650,12.1,3);
Segment_Array[3014].set_values(242112531,242112520,17.98,0,1650,1.9,3);
Segment_Array[3015].set_values(242112520,242112491,17.98,0,1650,1.3,3);
Segment_Array[3016].set_values(242112491,242112482,17.98,0,980,1.1,3);
Segment_Array[3017].set_values(242112482,242112476,17.98,0,980,0.2,3);
Segment_Array[3018].set_values(242112476,242112345,17.98,0,980,0.3,3);
Segment_Array[3019].set_values(242112345,242112335,17.98,0,980,7.3,3);
Segment_Array[3020].set_values(242112335,242112280,17.98,0,980,1.3,3);
Segment_Array[3021].set_values(242112280,242112279,17.98,0,9060,2.7,3);
Segment_Array[3022].set_values(242112279,242112256,17.98,0,9060,0,3);
Segment_Array[3023].set_values(242112256,242112231,17.98,0,9060,1,3);
Segment_Array[3024].set_values(242112231,242112228,17.98,0,9060,0.7,3);
Segment_Array[3025].set_values(242112228,242112216,17.98,0,9060,0,3);
Segment_Array[3026].set_values(242112216,242112215,17.98,0,9060,0.8,3);
Segment_Array[3027].set_values(242112215,242112213,17.98,0,9060,0.1,3);
Segment_Array[3028].set_values(242112213,242112204,17.98,0,9060,0.1,3);
Segment_Array[3029].set_values(242112204,242112118,17.98,0,9060,0.3,3);
Segment_Array[3030].set_values(242112118,242112165,17.98,0,9060,4.7,3);
Segment_Array[3031].set_values(242112165,242112166,17.98,0,9060,2.5,3);
Segment_Array[3032].set_values(242112166,242112157,17.98,0,9060,0.1,3);
Segment_Array[3033].set_values(242112157,242112129,17.98,0,9060,0.5,3);
Segment_Array[3034].set_values(242112129,242111981,17.98,0,9060,1.3,3);
Segment_Array[3035].set_values(242111981,242111734,17.98,0,9060,11.2,3);
Segment_Array[3036].set_values(242111734,522111573,17.98,0,250,18.5,3);
Segment_Array[3037].set_values(522111573,102111464,17.98,0,250,18.4,3);
Segment_Array[3038].set_values(102111464,102111450,17.98,0,1170,10,3);
Segment_Array[3039].set_values(102111450,102111443,17.98,0,1170,2,3);
Segment_Array[3040].set_values(102111443,522111200,17.98,0,1170,0.2,3);
Segment_Array[3041].set_values(522111200,422111179,17.98,0,1170,9,3);
Segment_Array[3042].set_values(422111179,422110995,17.98,0,575,0.7,3);
Segment_Array[3043].set_values(422110995,422110962,17.98,0,575,12.7,3);
Segment_Array[3044].set_values(422110962,422110960,17.98,0,11230,1.9,3);
Segment_Array[3045].set_values(422110960,422110955,17.98,0,11230,0.2,3);
Segment_Array[3046].set_values(422110955,422110959,17.98,0,11230,0.5,3);
Segment_Array[3047].set_values(422110959,422110981,17.98,0,11230,0.2,3);
Segment_Array[3048].set_values(422110981,422111016,17.98,0,11230,0.7,3);
Segment_Array[3049].set_values(422111016,422110943,17.98,0,11230,1.3,3);
Segment_Array[3050].set_values(422110943,422110931,17.98,0,11230,2.3,3);
Segment_Array[3051].set_values(422110931,422110931,17.98,0,11230,0.1,3);
```

```
Route *Route_Array;
Route_Array = new Route[50];
```

```
int route0[73]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
                23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,
```

```

43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,
63,64,65,66,67,68,69,70,71,72};

int route1[74]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,
44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,1063,
1064,1065,1066,1067,1068,1069,1070,1071,1072,1073};

int route2[79]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,
44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
64,65,66,67,68,69,70,71,77,74,75,76,3075,3074,3073};

int route3[52]={78,79,3002,3003,3004,3005,3006,3007,3008,3009,3010,3011,3012,
3013,3014,3015,3016,3017,3018,3019,3020,3021,3022,3023,3024,
3025,3026,3027,3028,3029,3030,3031,3032,3033,3034,3035,3036,
3037,3038,3039,3040,3041,3042,3043,3044,3045,3046,3047,3048,
3049,3050,3051};

int numRoutes = 4; //number of routes to be analyzed
Route_Array[0].set_values(0, 73, route0);

Route_Array[1].set_values(1, 73, route1);

Route_Array[2].set_values(2, 79, route2);

Route_Array[3].set_values(3, 52, route3);

int hsl, cars;
double cost_alternate = 0;
double cost_injury = 30000;
double cost_life = 3700000;

double killed, p_hit, p_k_re, pop_exp;
double p_re_hit = 1.0/3; //probability of release given that a car is hit

double tons_Cl_car = 90.0;
double grams_Cl_per_ton = 907184.74;
double L50 = 210; //the dosage exposure of chlorine for 30 min for which
//50% of the population dies
double sigma = 420 / pow(2*M_PI,0.5); //the standard deviation of the death
//rate from chlorine exposure for 30 minutes
double percent_release = 0.50; //amount of chlorine release if an impacted
//tank car ruptures
double radius = 800.0;

```

```

fp_out << "Route#,#cars,w_haul, w_death, w_injury, w_cleanup,totalHaulCost,"
        << "totalDistance,totalLivesLost,totalCostofLife," <<
        "totalInjured,totalCostInjured,totalVolumeReleased," <<
        "totalCostCleanup,maxLivesLost,maxInjured," <<
        "maxVolReleased(stons),maxCostperSeg,Risk Score" << endl <<endl;

for (int i = 0; i < numRoutes; i++) //loop over routes
{
    for (cars=1; cars<=10; cars ++)
    {

        //now the program varies the weights
        for (int w = 0; w < numWeights; w++)
        {
            bool doneVaryingWeight = false;
            bool doneWithDescent = false;
            bool setToOriginal = false;
            while (!doneVaryingWeight)
            {
                //the below code is to avoid strange representation of zero
                for (int a = 0; a < numWeights; a++)
                {
                    if (weightArray[a] < 0) weightArray[a] = 0.0;
                }

                fp_out << Route_Array[i].getRouteNum() << "," << cars
                    << ",";

                for (int z = 0; z < numWeights; z++)
                {
                    fp_out << weightArray[z] << ",";
                }

                double segScoreSum = 0.0;
                double totalHaulCost = 0.0;
                double totalDistance = 0.0;
                double totalLivesLost = 0.0;
                double totalCostofLife = 0.0;
                double totalInjured = 0.0;
                double totalCostInjured = 0.0;
                double totalVolumeReleased = 0.0;
                double totalCostCleanup = 0.0;
                double maxLivesLost = 0.0;
                double maxInjured = 0.0;
                double maxVolReleased = 0.0;
                double maxCostperSeg = 0.0;
            }
        }
    }
}

```

```

for (int j = 0; j < Route_Array[i].getNumSegs(); j++)
{
    //begin calculations for route segment
    int segArrayNum = Route_Array[i].getSegNum(j);
    double volume_hemisphere = 2*M_PI*pow(radius,3)/(3*0.001);
    //the above is the volume of the hemisphere of Cl2
    //around the release point site
    double Ldoes = p_re_hit * cars * tons_Cl_car * percent_release *
        grams_Cl_per_ton / 3.2 * 1000000 /
        volume_hemisphere;
    double distMeters = Segment_Array[segArrayNum].getDistMeters();
    if (distMeters <= 1600) distMeters = 1; //prevent divide by zero
    p_k_re = 1/(1+exp(-(Ldoes - L50)/sigma));
    pop_exp = Segment_Array[segArrayNum].getPopSeg() * 320000
        / 2589988.11;
    //1600m * 1600m = 320000 m^2
    //1 square miles = 2 589 988.11 square meters (says google)
    killed = p_k_re * pop_exp;
    double injured = pop_exp - killed;
    double volume_released = p_re_hit * 90 * percent_release * cars;
    //tons Cl2 released

    double cost_cleanup = log(volume_released) * 18000;

    double seg_cost = weightArray[0] *
        Segment_Array[segArrayNum].getCost_CarMile() *
        cars *
        Segment_Array[segArrayNum].getDistMiles() +
        weightArray[1] * killed * cost_life +
        weightArray[2] * injured * cost_injury +
        weightArray[3] * cost_cleanup;
    totalHaulCost += Segment_Array[segArrayNum].getCost_CarMile() *
        cars *
        Segment_Array[segArrayNum].getDistMiles();
    totalDistance += Segment_Array[segArrayNum].getDistMiles();
    totalLivesLost += killed;
    totalCostofLife += killed * cost_life;
    totalInjured += injured;
    totalCostInjured += injured * cost_injury;
    totalVolumeReleased += volume_released;
    totalCostCleanup += cost_cleanup;
    if (totalLivesLost > maxLivesLost) maxLivesLost = killed;
    if (injured > maxInjured) maxInjured = injured;
    if (volume_released > maxVolReleased)
        maxVolReleased = volume_released;
    if (seg_cost > maxCostperSeg) maxCostperSeg = seg_cost;
}

```

```

    segScoreSum += seg_cost;
} //end segment loop

//adjust weights
if (weightArray[w] <= 0.0)
{
    doneWithDescent = true;
}
if (!doneWithDescent)
{
    //adjust variable weights
    weightArray[w] -= 0.01;
    double sumVariableWeights = 0.0;
    for (int a = 0; a < numWeights; a++)
    {
        if (a != w) //if the weight we are looking at is not equal to
            //the fixed weight
        {
            sumVariableWeights += weightArray[a];
        }
    } //end finding sum of the variable weights
    for (int a = 0; a < numWeights; a++)
    {
        if (a != w)
        {
            //find ratio
            double ratio = weightArray[a]/sumVariableWeights;
            double increaseAmount = ratio * 0.01;
            weightArray[a] += increaseAmount;
        }
    }
}
else
{
    if (!setToOriginal)
    {
        setToOriginal = true;
        for (int i = 0; i < numWeights; i++)
        {
            weightArray[i] = originalWeightArray[i];
        }
    } //end if

    if (weightArray[w] >= 1.0) doneVaryingWeight = true;
    if ((!doneVaryingWeight) && (setToOriginal))
    {

```

```

weightArray[w] += 0.01;
double sumVariableWeights = 0.0;
for (int a = 0; a < numWeights; a++)
{
    if (a != w) //if the weight we are looking at is not equal to
                //the fixed weight
    {
        sumVariableWeights += weightArray[a];
    }
} //end finding sum of the variable weights
for (int a = 0; a < numWeights; a++)
{
    if (a != w)
    {
        //find ratio
        double ratio = weightArray[a]/sumVariableWeights;
        double decreaseAmount = ratio * 0.01;
        weightArray[a] -= decreaseAmount;
    }
} //end if(!doneVaryingWeight)
else
if (doneVaryingWeight && setToOriginal)
{
    //set to original weights
    for (int i = 0; i < numWeights; i++)
    {
        weightArray[i] = originalWeightArray[i];
    }
} //end else
fp_out << totalHaulCost << "," << totalDistance << "," <<
    totalLivesLost << "," << totalCostofLife << "," <<
    totalInjured << "," << totalCostInjured << "," <<
    totalVolumeReleased << "," <<
    totalCostCleanup << "," << maxLivesLost << "," <<
    maxInjured << "," << maxVolReleased << "," <<
    maxCostperSeg << "," << segScoreSum << endl;

} //done with varying one weight (while loop)
} //done selecting individual weight to vary (for loop)
} //end cars loop
} //end route loop

cout << "-----Program done-----"
    << endl << endl;

```

```
fp_out.close();  
cin >> sillyVariableToKeepTheProgramWindowOpen;  
  
} // end of main program
```