APPLICATIONS OF GIS AND OPERATIONS
RESEARCH LOGISTICS PLANNING METHODS FOR
ARKANSAS RURAL TRANSPORTATION
EMERGENCY PLANNING

Principal Investigator: Dr. Manuel D. Rossetti
Co-Principal Investigator: Dr. Edward Pohl
Co-Principal Investigator: Dr. Fred Limp
Graduate Research Assistant: Jason Stout
Undergraduate Research Assistant: Douglas Marek
MBTC 2088
July 2008
TABLE OF CONTENTS

1. INTRODUCTION .................................................................................................................. 1

2. GEOGRAPHIC INFORMATION SYSTEMS ............................................................................. 5
   2.1. USING A GIS .................................................................................................................. 5
   2.2. BRIEF OVERVIEW OF COMMERCIAL SYSTEMS ...................................................... 7
       2.2.1. Additional Reading ............................................................................................... 9
   2.3. ISSUES IN USING GIS APPLICATIONS IN EMERGENCY PLANNING ..................... 9
       2.3.1. The GIS Interoperability Problem .......................................................................... 11
       2.3.2. Example Systems and Architectures ...................................................................... 13
       2.3.3. Functionality and Data Issues ............................................................................... 16

3. OPERATIONS RESEARCH METHODS IN EMERGENCY PLANNING ............................... 19
   3.1. OPTIMIZATION APPLICATIONS .................................................................................. 21
   3.2. SIMULATION APPLICATIONS ..................................................................................... 24

4. HAZARD ZONE DETERMINATION VIA OPTIMIZATION ..................................................... 36
   4.1. BACKGROUND AND LITERATURE ON THE PROBLEM ......................................... 38
   4.2. FORMULATION OF OPTIMIZATION MODEL .......................................................... 41
       4.2.1. The Cova-Church Heuristic .................................................................................... 44
       4.2.2. Example Operation of the Cova-Church Heuristic ............................................... 47
       4.2.3. Genetic Algorithm ................................................................................................. 54
   4.3. TESTING AND EVALUATION METHODOLOGY ....................................................... 56
       4.3.1. Problem Instance Generation ................................................................................ 56
       4.3.2. Algorithm Implementations .................................................................................. 59
       4.3.3. Algorithm Evaluation and Experimental Design ................................................. 69
       4.3.4. Evaluation Results ................................................................................................. 71
   4.4. APPLICATION ON A REAL PROBLEM ...................................................................... 101

5. SUMMARY AND CONCLUSIONS ....................................................................................... 105

6. BIBLIOGRAPHY ................................................................................................................ 111
ABSTRACT

When a disaster strikes, emergency planners need to be prepared to handle many important duties, such as directing evacuations and distributing emergency supplies. Therefore, emergency planners rely on decision support systems (DSS), which help them to carry out these duties. To be most effective, a DSS should be integrated with a geographic information system (GIS), which provides analysis for the problems that arise and helps users of the DSS to visualize the situation. In addition, an effective DSS should include simulation models and optimization techniques, especially in the pre-event planning. For instance, a simulation model could be used to determine the fastest method of evacuation so that when a hurricane approaches an area, police can use this method to appropriately direct traffic. This report explores the many issues involved in integrating a GIS into a DSS, creating simulation models, and applying optimization techniques so that emergency planners are well prepared should a disaster occur.

To better examine the use of optimization in emergency planning this report investigates the determination of hazard zones within a geographic area. A hazard zone is an area corresponding to a worst-case evacuation scenario and is a potential vulnerability that should be addressed by risk mitigation strategies. The research compares two heuristic approaches to determining hazard zones based on population and road connectivity within a spatial network. The genetic algorithm and the Cova heuristic both have strengths and weaknesses as discussed throughout this report. The Cova heuristic performs better in sparsely connected networks with smaller hazard zones, while the genetic algorithm performs better in terms of time to obtain a solution and the quality of the solution when the network is dense and has larger hazard zones. Based on these results, the Cova heuristic is recommended for sparse rural networks.
1. INTRODUCTION

Emergency evacuation planning is of critical importance to our nation’s response to both man-made (e.g. terrorist attack) and natural disasters. The very essence of a disaster is what makes them hard to overcome. Disasters can be described by the following five major characteristics: 1) disasters are large, rapid-onset incidents relative to the size and resources of an affected jurisdiction; 2) disasters are uncertain with respect to both their occurrences and their outcomes; 3) risk and benefits are difficult to assess and compare; 4) disasters are dynamic events; 5) disasters are relatively rare (National Research Council, 2007). Because of the unpredictable nature of disaster events, emergency management is generally a relatively difficult task. If emergency planners can be given better tools for mitigation and evacuation planning, the chances of survival of people affected by the hazardous event will be greater. Tools that emergency planners can utilize are broadly defined as decision support systems (DSS).

Decision support systems are broadly defined computer-based information systems that support decision-making activities. For one to understand the implications of DSS in emergency management, one must first understand the disaster and emergency planning cycle (Johnson, 1992). The typical emergency management cycle consists of four main phases: preparedness planning, response, recovery, and mitigation. The preparedness planning phase consists of the activities undergone in the short term prior to the disaster strike that enhances the readiness for effective response such as hazard analysis and vulnerability analysis (National Research Council, 2007). The response phase consists of the myriad of activities undertaken immediately following a disaster to provide emergency assistance to the victims of the disaster (National Research Council, 2007). The recovery phase consists of both short-term and long-term activities undergone to return the people and property in the affected area to at least their pre-
disaster condition (National Research Council, 2007). Finally, the mitigation phase consists of the long-term activities undertaken after a disaster to prevent emergencies and to reduce damage resulting from those emergencies (National Research Council, 2007).

Decision support systems for emergency management often require a geographic information system component. A geographic information system (GIS) is a computer-based software tool that facilitates the analysis and mapping of information within a geographical area. The potential of integrating DSS with GIS and emergency planning tools has yet to be fully realized. Geospatial data, which describes the location of objects on the Earth’s surface, should and must be an essential part of all aspects of emergency management.

Although it is widely acknowledged that maps are essential in the earliest stages of search and rescue, the necessary investments in resources, training, and coordination are rarely given sufficient priority either by the general public or society’s leaders (National Research Council, 2007). For geospatial data integrated with appropriate management tools, there is a great potential to contribute to the saving of lives, the limitation of damage, and the reduction in the costs to society of dealing with emergencies. Natural disasters and catastrophes can have the potential magnitudes to overwhelm an agency’s resources as in the case of Hurricane Katrina during the summer of 2005, which claimed the lives of at least 1,836 while causing confusion and shock throughout the nation (National Research Council, 2007).

Geographic information system technology brings the ability to integrate, store, process, and output geographic information in order to assist the decision maker in making correct decisions in an emergency situation such as an earthquake. Through the aid of GIS technology, emergency managers can achieve a variety of efficiencies and gains in productivity, which include efficiencies gained from automation and the
efficiencies obtained when GIS performs tasks that are too time consuming and too costly to be done manually. (Gunes, 2000)

Unfortunately, the use of geographic information systems, primarily for map analysis, is often not enough. The complex nature of emergency planning requires the integration of GIS technology with advanced optimization methods and with advanced simulation methods. The optimization methods typically found within the field of operations research require extensive datasets, which can be supplied by GIS technology. Optimization approaches allow for optimal planning and allocation of resource before and during disasters. Examples of the use of optimization models include logistic dispatching (Özdamar, 2004), allocation of people to safe areas (Tarabanis, 1999), and even evacuation difficulty for defining a worst case scenario for evacuation and determination of vulnerability (Cova, 1997), on which this document provides an extensive analysis. Advanced simulation methods are required because of the dynamic nature of disaster events and because of the uncertain nature of evacuation processes and procedures. An example of this is illustrated in the research conducted by de Silva et al. (2000).

Researchers in the area of emergency management planning have recognized the need for integrating data with analysis tools. In a special issue of the IEEE Transactions on Engineering Management, Tufekci and Wallace (1998) outlined the emergence of emergency management as an area where engineering tools and techniques should find more application. In particular, they advocated taking a systems perspective of emergency planning that couples the use of advance computer and communication technologies (wireless networks, automatic vehicle location, global positioning systems, geographic information systems, etc.) with advanced optimization and simulation techniques.
Emergency management professionals need to take advantage of this systems view in order to better understand the dynamic interactions associated with both pre-event and post-event emergency planning. Pre-event planning involves the development of both mitigation/prevention plans and disaster reaction planning. Post-event emergency planning involves the dynamic execution and re-planning that must occur after an event occurs given the current state of the system. They argue that this holistic view will provide for a better understanding of the effects on life, property, society, and the environment from natural and man-made disasters. They state that, “emergency managers must have the ability to predict and analyze potential danger and develop necessary action plans for mitigating the conditions and responding to them when the inevitable takes place”. The key to this ability is having the right information available at the right time to be used in the right tools for the analysis.

Motivated by this call for integration, this report overviews these three areas (geographic information systems, optimization methods, and simulation methods) in order to illustrate the potential applications of these techniques to emergency management planning. Section 2 provides a basic overview of geographic information systems. Sections 3 and 4 provide a review of literature detailing applications of optimization and simulation within emergency planning. Finally, Section 5 presents a detailed example of optimization methods applied to emergency evacuation risk planning. Thus, the information in this report should illustrate the feasibility and potential of integrating disparate information when planning emergency operations. Emergency planners can use this report to better understand how advanced technology is being used in the planning process. In addition, the report serves as a basis for future research in the area of decision support systems for rural transportation planning in emergency response situations.
2. GEOGRAPHIC INFORMATION SYSTEMS

A geographic information system (GIS) is a computer-based software tool that facilitates the mapping and analysis of information within a geographical area. It has similar functions as a map but with the extensive features that increase its flexibility, speed and ease of use because of its ability to perform statistical analysis, geographic analysis or the analysis of vehicle routes. Although mapmaking and geographic analysis can be performed via manual methods, it is far easier and faster using GIS.

There are two primary types of geographic models used in the geographic information systems: the vector model and the raster model. The vector model is designed to store and encode information as a collection of coordinates. For example, it describes the position of a bore hole as a point with single coordinate, while the position of the river or road can be encoded as a linear feature and stored as a collection of point coordinates. Areas, for example, sales territories, may be recorded as a closed loop of coordinates. The vector model is only especially useful for describing discrete and static geographic features. The raster model can describe continuous varying features such as the accessibility costs for hospitals or the soil type. The raster model will encode the image into a collection of multiple grid cells.

2.1. Using a GIS

There are five basic tasks involved when using GIS: input, manipulation, database query and analysis, and visualization. During the input process, geographic data in a GIS compatible format will be entered into the system. There are various types of geographic data that can be obtained from the data suppliers, including base maps, business maps and data, environmental maps and data, and general reference maps. Base maps provide information regarding the multilevel road network; the boundaries of city and state; major
lakes and rivers; U.S. street and address data; postal code; and USGS raster maps. Business maps and data will often include statistical records related to census/demography, transportation, telecommunications, financial services, consumer products, and emergency facilities. Environmental maps and data are designed for analysis of weather, satellite imagery, environmental risk, and natural resources. General reference maps are used as the foundation for the database by showing the world and country maps. Most GIS data can be obtained from the Internet, varying in cost from free to thousands of dollars. A number of popular data sites include:

- GIS Data Depot (http://data.geocomm.com/)
- Geography Network (http://www.geographynetwork.com/data/index.html)
- ESRI datasets (http://www.esri.com/data/index.html)
- The National Map by USGS (http://nationalmap.gov/)

Once the basic input process has been completed, data manipulation is typically the next step. In this process, the geographic information (the map data) will be transformed to the same scale as the GIS system. This transformation can be temporary for display purposes or permanently for analysis. For example, most of the map data are available in different levels of detail or accuracy; therefore, it will have to be manipulated into the same level of detail for the analysis. For example, less detailed census boundaries to more detailed street centerline files may need to be modified. This process may also entail coordinate transformations so that all datasets are operating within the same coordinate system.

During the database query and analysis task, the GIS is utilized as a database management system to store, organize, manage, and query data. Most systems present the data in the form of tables via a relational design. With the relation design, data can be
stored in a collection of tables and these tables will use to link the common fields together. During the query and analysis process, the user has the ability to ask some analytical questions such as:

- Where is the best location for starting a new business?
- Where is the hazardous zone?
- Will it affect the traffic by building a highway here?

Most of the questions asked are based on “what if” scenarios. Proximity analysis and overlay analysis are two very important GIS analytical tools often used during this process. In proximity analysis, a buffering process is used to determine the proximity relationship between features. For example, a proximity query can be performed to obtain the approximate number of the houses that lie within a certain area. In an overlay analysis, the different data layers will be joined or integrated together physically. For instance in an overlay analysis, the data on soils, slope, and vegetation, or land ownership with tax assessment can be joined together.

The last step is often the visualization process. In this process, most of the end results and analysis will be displayed in map form. However, those maps can be transformed into report format by displaying statistical graphs, charts, photographic images, and other output.

2.2. Brief Overview of Commercial Systems

There are a number of commercial GIS products that are popular. A search of the Internet can provide a basic overview and reviews of the available software offerings, for example: ArcGIS, Intergraph, Idrisi, MapInfo, and GRASS. This report provides a basic
discussion of ArcGIS. The discussion focuses on the basic functionalities of the system to provide a context for further discussion during the literature review portions of this document.

ESRI ArcGIS Desktop, consists of three versions: ArcView, ArcEditor, and ArcInfo. The difference between the versions is that ArcEditor has more tools and applications than ArcView, and ArcInfo has more tools and applications than ArcEditor. ArcView can perform mapping, spatial analysis, database integration, and comes with ready to use datasets. ArcEditor allows for editing and managing geographic data. It essentially does everything that ArcView does and enhances data validity and manipulation. ArcInfo does everything that ArcEditor does plus allows for advanced spatial analysis (e.g. proximity analysis, overlays, surface analysis, etc.), allows the creation of databases and schemas, and permits high-end cartography. The similarity between the three versions is that they have the same three applications: ArcMap, ArcCatalog, and ArcToolbox. A listing of their functional coverage can be found at:


In ArcMap, users make maps using layers of spatial data and colors and symbols chosen as well as the ability to query attributes, and analyze spatial relationships. In ArcCatalog, spatial data can be viewed and browsed through on a computer’s hard disk, on a network, or on the Internet. Spatial data can be searched, previewed, and added to ArcMap. Along with these features ArcCatalog also provides tools for creating and viewing information about spatial data, such as who created it, when it was created, what purpose it was created for and so forth also known as metadata. In ArcToolbox, tools are used to change the map projection of data and to convert spatial data from one format to another.

Additional reading on the basics of geographic information systems and their use in emergency planning can be found in the next sub-section.
2.2.1. Additional Reading

Pine (1998) provides a basic introduction to geographic information systems. It describes what a GIS is, the advantages and disadvantages of GIS, and geographic spatial concepts. The basic terminology used in GIS is presented in an introductory manner. In addition, the report summarizes the basic use of a GIS (spatial analysis, geo-coding, obtaining data/maps) and the basic hardware/software requirements of such systems. This report would be useful for emergency managers to better understand the basics of using GIS technology.

The white paper by Johnson (2000) provides an introductory review of the use of GIS technology in the area of emergency management. Like Pine (1998) this introductory report would be beneficial to emergency management professionals to understand the terminology and capabilities of geographic information systems. The paper provides a nice set of definitions for emergency management and reviews the basic phases of emergency management (planning, mitigation, preparedness, response, and recovery). These phases are then related to how a GIS can be utilized and the data requirements needed during that use.

2.3. Issues in Using GIS Applications in Emergency Planning

This section describes the application of GIS technology for emergency planning. The basic issues that emergency planners need to be aware of are discussed as well as a number of example applications. This should provide a good context for planners to better appreciate the challenges of applying this technology.

GIS has been used for many years in emergency planning situations. For example, Mondschein (1994) provides a review of the use of spatial information systems in emergency planning situations involving environmental impacts. After a brief review
of the environmental legislation that motivates the tracking and use of hazardous materials, the article provides a basic overview of how emergency responders can and should use the available data within geospatial databases. The author discusses surveys on the use of this technology by emergency responders and concludes that “emergency responders appear to be convinced of the importance of environmental information, although there is limited evidence on its actual use in emergencies.” The author also relates some of the challenges caused by the lack of standards in information collection and naming. The author suggests that ideal software should have “map support for spatial display of data, data management systems to support regulatory reporting and hazard identification, and plume dispersion modeling and communications to permit maps, data, and models to be transferred in a matter of seconds”.

In a more recent paper, Radke et al. (2000) discuss the ways that geographic information science (GIScience) can contribute to the area of emergency preparedness and response, especially in the areas of risk analysis, evaluation of preparedness, and information integration. The authors argue that researchers “must develop predictive and operational models that are embedded within geographic information systems.” Geographic information systems can play a major role in all phases of emergency planning; however, there are a number of significant challenges with respect to utilizing geographic information systems for emergency planning.

In Radke et al. (2000), the challenges to GIScience are elucidated within a context of a paradigm that divides emergency response along a three axis continuum consisting of types of hazards, the time associated with actions taken to the planning/response problem, and actions taken such as prevention, insurance settlement, etc. An overview as well as some literature review of the role of GIScience in natural disasters (earthquakes, volcanoes, tsunami, landslides, floods, tornados, and hurricanes) is given. The authors
then describe the role of GIScience in human induced hazards (epidemics, war, spills, explosions, and fires). One of the key enablers for addressing problems in these areas that are facilitated by GIScience is the potential for data acquisition and integration.

A key conclusion based on their analysis is that the “overall research challenge in spatial data acquisition and integration for emergency management can be viewed as one of delivering accurate, appropriate information to all parties involved at the proper stages of the disaster in a timely manner.” The authors describe some of the research questions to help with this area: distributed computing, extending geographic representations (e.g. how to represent risk and human vulnerability), and interoperability of geographic information, human computer interaction, application scale, spatial analysis, and uncertainty representations in GIS data. Application scale refers to the issue of the appropriate data modeling scale for problem analysis. For example, when performing a simulation how does one integrate meso-simulation analysis with macro-level analysis. The authors wrap up their analysis with a discussion of the implications for GIScience education and issues related to public policy and its interaction with GIScience. A comprehensive list of references is provided.

A major point in Radke et al. (2000) is the interoperability of data, which concerns the ability of two or more systems to share data and tools effectively.

2.3.1. The GIS Interoperability Problem

The 2002 Los Alamos National Laboratory report by Keating et al. summarizes the many challenges faced by organizations to effectively share GIS data across organizational boundaries, especially within the context of emergency management planning and response.
As is common in most of the literature in this area, the report suggests that a particular architecture, Enterprise GIS, is and will be a key component of any solution to this problem. Enterprise GIS connotes the concept of totally a integrated system at the highest level (enterprise) capable of serving all geospatial needs (source, database, and applications). Within the context of an emergency response for the Cerro Grande Fire, the Keating et al. (2002) motivate the complexity of the data requirements, the groups/organizations that need to be involved, and the application of GIS. They describe some of the challenges to integrating data at this level and enumerate a number of requirements for data warehouses to support the effort. They conclude that the “key to success for enterprise GIS at LANL is the development of a sound geospatial information management plan.” That is, all stakeholders should be involved in the management of the necessary information. This is very crucial for prompt and appropriate emergency response.

For the issue of interoperability and data sharing to be understood, the two basic levels of interoperability, technical and semantic, must be understood (National Research Council, 2007). Technical interoperability concerns the fundamental building blocks of the software and hardware, which generally involves issues with format; however, problems with technical interoperability are typically the easiest to solve even though different data formats encountered in the field during response can cause significant delays. “Technical interoperability can typically be achieved by selecting and implementing the appropriate software and Internet standards, common content encodings for transmission, and so forth (National Research Council, 2007).”

Semantic interoperability, which concerns differences in concepts and meaning given to data by different users and systems, is often more problematic than technical interoperability. Core semantic differences arise from the selection and definition of
keywords or technical terms used within the system. There is an enormous variety of ways in which geospatial data is encoded with a large number of classification schemes, vocabularies, terms, thesauri, and data definitions in use by different data-producing agencies causing semantic interoperability to be relatively challenging. In basic principle, metadata, or data about data, provides the basic foundation for semantic interoperability by defining the meaning of each of terms that underlie the data production process (National Research Council, 2007).

Gunther and Voisare (1998) describe the information within geographic information systems and its representation. Typical uses of metadata include, supporting efficient indexing, browsing and navigation, specifying uncertainty in the data, and unifying relationships between disparate data sources. The authors describe the fundamental types of metadata and how it is being used. A better understanding of metadata in the GIS community is necessary to facilitate the goals of the U. S. Government’s National Spatial Data Infrastructure initiative. The authors describe this initiative its standards and their relationship/dependence on metadata. The authors also review efforts within the European Union for investigating metadata standards for spatial and environmental data. They conclude that metadata is improving the availability and quality of geographic information with more work necessary on architectures that can take advantage of such information. Overall, issues related to interoperability can be typically addressed through the establishment of appropriate standards (National Research Council, 2007).

2.3.2. Example Systems and Architectures

A large amount of effort has been expended to develop systems and architectures that attempt to integrate data, especially in the emergency management domain. For example,
Zlatanova and Holweg (2004) propose a software architecture for managing geo-information in emergency response, especially for the case of real-time response. The architecture is described from the view of end-users, the middleware needed, and the database management. They suggest that emergency situations involving flooding, terrorist attack, and fire could be managed by such a system. They present general requirements for such a system and postulate the kinds of data, visualization, and integration that would be needed. They conclude that such systems will be increasingly needed to improve disaster management.

Iakovou and Douligeris (2001) report on the architecture, development, and functionality of IMASH (an Information Management System for Hurricane disasters). The research is motivated by the need to integrate all the disparate information requirements that occur during a hurricane: geographical, ecological, facilities, meteorological, etc. in order to effectively develop and execute evacuation contingency plans. The IMASH system provides a graphical user interface based on GIS capabilities, real-time communications, and an object-oriented database system. The authors review the plethora of systems that are available to emergency responders, and in particular the Oil Spill Information Management System (OSIMS) by Douligeris et al. (1995). The IMASH system is based on a distributed architecture involving the Generic Mapping Tool, Advanced Visualization System, the object-relational system, Oyster, and Java. The paper also provides an overview of the modules and their hierarchical arrangement within the system. Since the system was a prototype, no specific results on the use of the system were presented.

Lurie et al. (2002) describe GeoView, a software toolkit for dynamically displaying GIS information, especially on a web-based appliance. A key finding in this research effort is that it recognizes the need for the dynamic display of GIS information,
in lieu of the traditional display of static GIS products. The paper describes the software architecture, especially its object-oriented design and its layers of abstraction. The paper also describes a number of applications in which GeoViewer was a key enabler: Object-oriented integrated landscape analysis and modeling system (OO-IDLAMS), Joint Warfare System (JWARS), and Global Theater Weather Analysis and Prediction System (GTWAPS). Such an architecture would be very useful in integrating a simulation representation with a geographical information system when performing an emergency evacuation study.

While many systems focus on data/system interoperability, the system by Rauschert et al. (2002) is focused on how to make the people involved in using an GIS more productive. They contend that GIS software has been inherently oriented towards single-user use and do not facilitate the type of collaborative decision-making required in emergency management. They call for user interfaces that “support the ways in which humans work and interact in a collaborative emergency task situation” requiring “multimodal, rather than unimodal, collaborative rather than personal, and dialogue-enabled, rather than unidirectional.” They present a system/interface that uses speech and gesturing integrated with GIS called DAVE_G (Dialogue Assisted Visual Environment for Geoinformation). They review other work on human computer interaction to motivate the requirements for DAVE_G and provide an overview of the basic architecture and functionalities of a prototype system. Cai et al. (2006) describe the architecture and implementation of DAVE_G. They also compare and contrast DAVE_G’s functionality to that of a desktop GIS. This follow up work to Rauschert (2002) overviews the need for better collaboration between entities involved in geospatial analysis and the development of tools to improve this collaboration. In particular, they describe a prototype system called (HI-SPACE) human information workspace, which
concentrates on improving the use of computing displays of spatial information when working in a group. The system is based on a neural network that has been trained to recognize human gestures.

2.3.3. Functionality and Data Issues

Interoperability is essentially about the ability to efficiently and effectively share data. Geospatial data can be characterized as three basic types: framework data, foundation data, and event-related data. Framework data comprise the geographic themes of geodetic control (system of measurements used to establish the shape of the Earth and to lay out its basic coordinate system), orthoimagery (high-resolution images derived from aerial photographs or satellites), elevation, transportation, hydrography (locations and properties of bodies of water), governmental units (locations and properties of administrative areas), and cadastral information (map of land ownership with property boundaries). Foundation data relate to a specific organization’s mandate and is, therefore, related to the framework data; however, foundation data includes information pertaining to items such as underground pipes, overhead power lines, etc. Lastly, event-related data corresponds to information collected specifically pertaining to particular events, such as location of casualties, response resources, and inventories of environmental damage.

Once the data has been effectively managed, the GIS can be used to perform an analysis on the data. Some of the basic capabilities or functions of typical GIS programs relevant to emergency management are: address matching or geocoding, routing and allocation, Thiessen polygon or Voronoi diagrams, buffering, dynamic segmentation, and overlay functions (Parentela, 2000).
Geocoding refers to software providing spatial coordinates to objects with address matching being the process of matching an object with its correct spatial address. Geocoding is a four-step process that includes the preparation of a reference theme or base map for geocoding, creating an address table, matching the addresses, and creating geocoded points. Examples of geocoding strategies include latitude/longitude, x/y coordinates, and street addresses. Routing tools facilitate finding a route or path within a network between a given origin and a destination. Allocation is the process that is used to identify the maximum coverage that can be achieved from a location given the constraints of that location and is typically very effective when evaluating response times of emergency response. Buffering is a common method of reclassification. Through the buffer command, a polygon is created around an object with the region inside the polygon representing the area that is within the tolerance specified by the buffer.

A Thiessen polygon or Voronoi diagram is used to determine the influence of a point data or center to its proximate region. Estimates of the service areas of emergency responders can be obtained using a Thiessen polygon, which would find the proximal region of influence to be served by the responder or point. Lastly, dynamic segmentation is a method of evaluating and combining multiple sets of linear attributes, such as a route network where events are then related to the route system where the segments are not predefined. Dynamic segmentation can typically aid in risk estimation and identifying critical route segments for emergency response planning. (Parentela, 2000)

Since emergency management planners must utilize available data for contingency analysis, the data that is available is crucial for appropriate plans to be developed. For example, for earthquake disaster planning the following data would be extremely useful (National Research Council, 2007):
• Locations of population and critical infrastructures, including bridges, overpasses, dams and reservoirs, aqueducts, day care facilities, hospitals, schools, nursing homes, and urgent care facilities with up-to-date occupancy data; locations of critical infrastructure potentially impacted by the event; shelter locations and capacity, and service areas

• Location of transportation routes, which includes road networks and capacity, traffic flows based on time of day, egress routes, and traffic control points for the main state and county roads

• The location of special-needs residents; hospital capacities; the number and locations of households without private automobiles, which is critical for evacuation planning; the availability, capacity, and locations of public transportation such as school buses, which could assist in evacuation

• Presence of pets in a household, which is a key determinant of a resident’s willingness to comply with evacuation orders

• Basic information on numbers of people and numbers of households, age characteristics, race ethnicity, and language spoken available at the block level for a city and county

• Locations of potable water lines, wastewater treatment, facilities, sewer lines, natural gas lines, oil storage facilities, electric lines, communication lines, and cell phone towers

• Building inventory data; locations of hazardous materials treatment, storage, and disposal facilities and exact type and quantities of materials at each location

• Daytime as well as nighttime population data

• Location of active faults and the strength potential of the fault area should be recorded (Johnson, 1992).
Within this context mitigation measures would include data gathering, mapping and analysis of earthquake-related phenomena, avoiding construction on faults, minimizing population and housing densities near faults, strengthening building codes, and public acquisition of land (Johnson, 1992).

As noted in Radke et al. (2000) and Tufekci and Wallace (1998), it is not just the geographic information system and its technology that is the key to its application in emergency planning but rather how it facilitates the development and use of predictive and operational models. The next section overviews some applications of operations research models within this context.

3. OPERATIONS RESEARCH METHODS IN EMERGENCY PLANNING
Sparse research has been conducted in the development of emergency planning tools through integration with optimization models and/or simulation models.

Optimization models are mathematical constructions or representations of systems that strive for the ultimate goal of determining the globally best solution or solutions for the constrained system. These models are largely prescriptive in nature, recommending a solution or making an optimal decision. Simulation models provide a dynamic, descriptive form of modeling to enable the understanding of the behavior of the system under a wide-variety of complex parameter configurations. Simulation models used in emergency planning consist of three basic types: micro-simulators, macro-simulators, and meso-simulators (Pidd, 1996). Micro-simulators attempt to track the detailed behavior of individual entities in the simulated situation; whereas, macro-simulators make no attempt to track the detailed behavior of individual entities (Pidd, 1996 and de Silva, 2000). Meso-simulators are a basic compromise between micro- and macro-simulators that usually involves discrete simulation that tracks the behavior of groups of entities (Pidd, 1996 and de Silva, 2000).
Integration of GIS with either an optimization or simulation model is typically achieved through two coupling methods, which are loose and deep coupling. Loose coupling facilitates integration through a file exchange mechanism (Gomes and Lin, 2002). Typical loosely coupled integration of GIS with some optimization model involves writing solutions and data to ACII files, then using the GIS to perform spatial analysis with the solutions being generated from the optimization model or search heuristics (Ducheyne, 2006). Deep or tight coupling uses multiple criteria evaluation functions fully integrated with the GIS, a shared database and a common user interface (Gomes and Lin, 2002).

A prime example of deep coupling is in the research of Wang (2005), which integrates a simulation model with GIS. Wang (2005) presents the benefits and challenges of integrating three components of information technology: a GIS, simulation models, and a 3D visualization. Although a few such integrated systems exist, the development of each of these components has mostly been independent. One challenge has been the coupling of a GIS and simulation models, especially in terms of sharing information. In this approach to deep coupling, a single user interface controls both the GIS and the simulation models, even if they remain separate systems. Another challenge has been visualizing the results after the GIS and simulation models have been integrated. Typically, when a GIS data file is converted into a format that can be visualized, only the geometry is retained. This research describes a graphic user interface, which was constructed in Visual Basic, and was used to visualize the simulation within the spatial context. Although the GIS and simulation model were integrated through a deep coupling method, the simulation model and GIS still remain separate systems (Wang, 2005).
The following two sections present an overview of various applications of optimization and simulation within the context of emergency planning, especially with respect to integration with a geographic information system.

3.1. Optimization Applications

In the literature, there are a variety of optimization models and methods that are integrated with GIS, such as a linear and integer multi-period multi-commodity network flow model (Özdamar, 2004), multiple objective genetic algorithms (Ducheyne, 2006), multi-objective linear programs (Gomes and Lin, 2002), and a dynamic allocation model for people to “safe zones” (Tarabonis, 1999). Wright et al. (2006) discuss an extensive amount of literature relating to the applications of various algorithms, models, and techniques towards the goals of homeland security for the United States. Wright et al. (2006) develop a two-dimensional classification system for the emergency response literature. On one axis, literature is classified according to what type of possible disaster it relates to, such as chemicals, airline security, or cyber security. On the other axis, the literature is classified according its position in one of four stages of the disaster life cycle: planning, prevention, response, and recovery.

Gomes and Lin (2002) describe a general methodology for integration of GIS with multi-criteria methods for optimization purposes. The paper first discusses the basic differences between the loose coupling strategy, which facilitates integration with GIS through the use of an external exchange file, and the tight coupling strategy in which GIS is fully integrated with the multiple criteria evaluation functions. Some software packages such as IDRISI and SPRING have multi-criteria evaluation functions that use an Analytical Hierarchical Process (AHP) method. Gomes and Lin (2002) also discusses the methodology developed and apply it to a specific case study. The three stages of the
methodology are as follows: 1) in a GIS environment, the number of possible alternatives is reduced to only those that are feasible by physical and/or qualitative constraints, 2) a Multi-Objective Linear Programming (MOLP) is solved using the Pareto Race method through the use of VIG software, 3) the Pareto optimal results or efficient frontier of the MOLP are introduced into the GIS for visualization of the results.

Ozdamar et al. (2004) discuss the planning of logistics involved in dispatching commodities in emergency situations. A dynamic time-dependent transportation logistics model is developed in which new requests for materials become available during the current planning horizon. The model takes into account time-dependent supply, demand size, fleet size, and facilities’ schedules. The model is also a multi-period multi-commodity network flow problem with the possibility of split deliveries designed as a mixed integer linear program (MILP) that minimizes the sum of the unsatisfied demand of all commodities. Due to the computational issues associated with solving MILP problems, the problem is solved by splitting the model into an integer linear program (ILP) that deals with vehicles and a LP that deals with multi-commodity minimum cost network flow. A Lagrangean relaxation based heuristic approach is then used to couple the two sub-problems.

Saleh et al. (2007) discuss three different models of emergency evacuation route planning (EERP) for a natural disaster such as a hurricane. Each of the three models is viewed as a max flow problem that is solved using integer linear programming. The first evacuation model is a single-flow EERP with no contraflow. Contraflow refers to the reversal of lanes such that all the lanes of traffic on a road flow in one direction. This model considers the case where no action is taken for lane-reversals, which increases the capacity of the roads. The second evacuation model is a single-flow EERP with contraflow. In this model, lanes are reversed such that all traffic in both lanes are moving
in one direction. The third evacuation model is a two-flow EERP model with contraflow. In this model, roads use reverse-laning to increase capacity, but for certain periods, the roads switch direction such that emergency vehicles may travel the other direction toward the disaster area. The results of the research indicate that using contraflow decreases the evacuation time a modest amount, and using two flows to allow for the passage of emergency vehicles does not significantly impact the overall evacuation time.

Zografas et al. (1998) describe the development of a decision support system designed to assist electric utility companies ability to respond to disasters that cause power outages. In the pre-event analysis, utilities must design their operating network such that the reliability of the network is still maximized in the event of a disaster and the resources of the network can be allocated to minimize the disruption time in the event of an incident. Data for such as system included GIS information in the form of topographical maps, geometric and operational characteristics of the transportation network, boundaries of the service districts, natural and man-made barriers to emergency response teams, the locations of service calls, the locations of emergency response teams, and population characteristics. The types of decisions needed included the optimum partition of service areas, the optimum allocation of vehicles and teams to service areas, workload balancing with customer service considerations, and dispatching algorithms. The decision support system utilized this data to populate models for shortest path routing, traveling salesman problems, location/allocation decision-making, queuing analysis, and simulation. Clearly, such systems are data intensive and take specialized knowledge of both the domain and of the techniques. The system used a combined optimization and simulation approach to minimize service unavailability.

A number of other applications are included in Appendix A (Annotated Bibliography). As noted in Zografas et al. (1998), optimization is not the only concern of
many of these applications. Optimization excels at the pre-planning stage, to understand the constraints and decisions that can be set in advance. However, these plans must operate in a dynamic and stochastic environment. Simulation excels in its ability to understand the risks associated with these plans. This topic is discussed in the next section.

3.2. Simulation Applications

Since man-made or natural disasters frequently pose grave threats, the effectiveness of an evacuation event needs to be significantly improved in order to save human lives and property. While GIS provides the functions that help examine spatial relationships, simulation modeling is capable of representing the dynamic relationships between the cause and the effect (Wang, 2005).

Jain and McLean (2003) argue that because simulation can be used to model different aspects of an emergency system, there should be tools that allow for the integration of the various models. They propose a simulation framework that can help coordinate between the various simulation tools and integrate their modeling/planning capabilities. The authors provide a brief review of some tools to motivate the need for integration. Within the context of an example explosion at a commercial building, the authors identify the types of data that would need to be integrated: street maps, emergency facility locations, building structural drawings, building use information, traffic, hospitals locations, layout of nearby buildings, etc. A modeling of this situation might possibly entail a building and fire simulation model, traffic simulation models, command and control flow, and hospital simulation models. The authors suggest methods to integrate these models and motivate the need for the integration. They provide basic requirements for such a system in terms of a systems engineering analysis.
Wright et al. (2006) also discuss literature concerning the use of simulation during evacuations. Micro-simulation, which tracks the movement of individual entities such as people or vehicles, can be linked to a GIS to plan emergency evacuations. Also of use are macro-simulations, which approximate large-scale population flows as fluids, and meso-simulations, which track groups of individuals.

Fedra (1999) describes the relationships between environmental management and spatially distributed dynamic data. The author addresses the basic problem of static information in the form of databases and GIS and dynamic information via simulations, which may include a real-time element. While geographical information systems have excellent functionality with respect to spatial display of geo-referenced data, they are challenged to show the state of the system over time. This has implications with respect to analyses that require dynamics, such as environmental modeling involving mass or energy flowing through environmental mediums. The paper describes a number of applications/projects that have attempted this integration (ECOSIM, AIDAIR, SIMTRAP, and HITERM). These projects all “address tough problems: they are complex; dynamic, and involve large volumes of data as well as spatially distributed 3D phenomena and models” (Fedra, 1999). The author reviews the common architectures found in each of the applications, including client-server, real-time data acquisition via http protocols, multi-media user interfaces, geographical information systems, spatial capable simulation modeling, and expert system support. A unique feature of these efforts is the embedded capability to simulate the environmental process to provide “what if” impact assessment. For example, the author describes how a GIS and a simulation of traffic may allow for an analysis of street networks, fleet composition, accidents, and more to analyze the dispersion of emissions/air pollution. The author concludes with
some remarks involving how to transition these projects and their results to benefit public institutions.

Pidd et al. (1996) describe the creation of an early prototype of a spatial decision support system (SDSS), that is, a system that links a GIS to a simulation model. The purpose of this system was for planners of emergency evacuations to have immediate feedback for plans for various contingencies. The system, named configurable emergency management and planning simulator (CEMPS), proved that such a SDSS was very much possible. It is important to note that CEMPS was designed for contingency planning, not real-time emergency management. This early prototype of CEMPS used the GIS to carry out static analysis, such as the likely coverage of a plume of dangerous chemicals from an industrial accident. In addition, CEMPS had a micro-simulator that could simulate evacuations of vehicles. The simulator was written in C++ and extended existing classes, such as locations and arcs, to create classes for use in the simulator, such as junctions and roads.

De Silva et al. (2000) describe a more advanced version of CEMPS and explore the non-trivial issues involved in the creation of this version of CEMPS. The scenario used to test CEMPS was that of a large-scale road evacuation as a result of a nuclear disaster, and a simulation model that simulated the evacuees as they moved away from the danger zone towards a safe shelter. This version of CEMPS consists of four main parts. First, there is the simulation model that is capable of dynamic analysis and decision modeling. The second part is the GIS, which includes the spatial database and geographical analytical analysis tools. The third part consists of the integration link interface, which allows the GIS and simulation model to dynamically communicate and exchange data. The last part is the user interface that displays the current state of the simulated evacuation and allows the user to request information or run the simulation.
For its simulation needs, CEMPS uses a micro-simulator, which simulates the behavior of individual vehicles as they try to find the shortest route to the shelters given the prevailing traffic conditions. One important issue in this simulation is how to handle traffic generation. This involves making assumptions about how and when people will evacuate and developing a mechanism that approximates this behavior for the simulation. Another important issue is defining the goals of the evacuees being simulated. Either they need a specific shelter as their destination, or they need to have defined rules for a searching process such that at, for example, each road intersection, the evacuees can decide which way to go.

The GIS is a crucial part of CEMPS because it relates the simulation to the real world. The GIS also keeps a database of information concerning the road network, the evacuation area, and the geographical distribution of evacuees, and the user can retrieve all of this data as maps. Furthermore, GIS provides many of the analytical analysis and can provide information such as the number of evacuees in a certain area. For a large proportion of the research that has been done concerning the integration of GIS with simulation models, ARC/INFO is used for the various components that perform mapping and plotting functions, spatial data analysis, etc., along with a database system to store and handle the spatial data.

For the interface that links the GIS and the simulation, there are two important issues. First, there is the level of integration between the GIS and the simulation. CEMPS has a high level of integration, meaning the simulation and the GIS are able to run simultaneously, and they are able to exchange dynamic data with each other. The second important issue concerns the use of data formats and data exchange mechanisms. In order for the different parts of CEMP to communicate, they must use data formats that can be read by the other parts of the system. For CEMPS, this is usually standard ASCII.
text. When storing spatial data of a GIS, however, there are two different standards. The vector approach uses vector data, which has properties such as a direction, while the raster approach interprets data in terms of pixels. CEMPS uses the vector approach, and in order to use a raster approach based GIS, the data recognition procedures would have to be altered to be able to use both types of data.

While Pidd et al. (1996) integrate a discrete event simulation model with a GIS, recently, agent-based modeling and simulation has dominated this research area because it can capture individual and collective behaviors in a dynamic complex system. For example, Chen and Zhan (2008) used a simulation model to compare the evacuation times of simultaneous evacuations to the total evacuation times of staged evacuations where different zones are evacuated in a sequence at set time intervals. Basically, two types of evacuation strategies were investigated. One is a simultaneous evacuation strategy: all residents in the affected area are informed at the same time and then evacuate simultaneously. The other strategy is a staged evacuation: the affected area is divided into several different zones and residents in different zones are organized to evacuate in a specific order. The overall evacuation time of one evacuation event is calculated from the time of first agent starting to evacuate to the time of last agent leaving the affected area. This time typically represents the effectiveness of an evacuation event.

The evacuations are assumed to be conducted by vehicles along roads in a small urban area in response to a disaster such as a hurricane or wildfire. Three different road structures were tested: a grid structure, a ring structure, in which roads are either spokes emanating from the center of the city or different sizes of concentric circles, and a real road structure from the city of San Marcos, Texas. Several different population densities were used for each road structure. For the staged evacuations, the three road structures were each divided into four zones of equal size, and all the different combinations of
zone sequences were tested. The delay in evacuation from one zone to the next was one minute for the smaller population densities and four minutes for the larger population densities.

The simulation used agent-based methods, i.e. micro-simulation, meaning it tracked individual vehicles. An agent-based simulation tool called Paramics was used to model the traffic flows. By using Paramics, the researchers were able to access micro-level agent communication characterized by the interaction behavior of individual vehicles, and study the joint behaviors of the whole affected community. No traffic lights were used in the simulation, but at intersections a system was used in which straight traffic had right-of-way to traffic turning right which had right-of-way to traffic turning left. Dynamic routing was used, meaning that simulated drivers dynamically adjusted their route based on traffic conditions. The simulation used an advanced model governing the acceleration and deceleration of the vehicles based on factors such as the speed of and distance to the vehicle in front of it. In addition, the simulation included a factor of aggressiveness, which governed how fast a vehicle tended to drive, and the level of aggressiveness was normally distributed among the vehicles.

The results of the paper indicated that for low population densities, such as one or two vehicles per household, simultaneous evacuation was always the fastest. For large population densities, such as eight vehicles per household, the results were mixed. Simultaneous evacuation was still the fastest for the ring structure of city roads. For the grid structure and the actual structure of San Marcos, staged evacuations were faster when adjacent zones were not evacuated next to each other in the sequence. This was because when adjacent zones were evacuated at close to the same time they tended to use the same roads that quickly became congested, but non-adjacent zones did not have this problem.
However, there were a number of shortcomings in this model. First, the model assumed that any car reaching the edge of the affected area was automatically safe and disappeared, while a more realistic model might also include the traffic conditions outside of the affected area. Secondly, the model used only a small area, with a small number of households and roads being affected. If a large city or area was affected by a disaster on the scale of a hurricane or wildfire, there is little reason to believe the evacuation would be handled in the same way as a small city affected by a small disaster. Thirdly, only two time intervals, one minute and four minutes, and only one number of zones, four, were used in zone evacuation, although this was probably limited by the small size of the model. Finally, a traffic light system or police-directed traffic might be more realistic than the right-of-way method used. With these considerations, more research should be done before judging zone evacuations.

Wu et al. (2008) created a comprehensive simulation model for civilian emergencies and disasters. The model incorporates a GIS, databases, and several algorithms and optimizations to coordinate the response of emergency vehicles. The GIS contains geographically reference metadata such as roads, waterways, and topography. A discrete event simulation models the disaster incidents and the emergency response system is modeled as a transportation network, where important street intersections are network nodes and the connecting roads are arcs. Boats, helicopters, and trains are also used in the model, but in separate layers.

In Moynihan et al. (2008), a simulation model is developed which analyzes the movement of vehicles northbound on I-65 from Mobile, Alabama. The model keeps track of groups of 25 vehicles as they enter the highway, move through the highway, and exit. The model also keeps track of variables such as road length, number of lanes, average distance between vehicles, lane occupancy, distances traveled, evacuation rates,
average delays, and traveling speeds. The model examines the effect of “reverse-laning,” which is the reversing of one direction of traffic to increase the flow of traffic in the direction the vehicles are evacuating. The consequences of this method include restricting the flow of emergency vehicles traveling the other direction. The model also examines the effect of closing several on-ramps on the overall effectiveness of the evacuation.

Usher and Tate (2007) discuss the development of a simulation system that models the behavior of pedestrians. The purpose of the paper was to help in the design of important places with pedestrian traffic, such as buildings, courtyards, and convention centers, and also to help in the planning of emergency evacuations.

Usher and Strawderman (2008) continue the work described in Usher and Tate (2007) and discusses how a micro-simulator is used to model pedestrian traffic. The simulator models a pedestrian in 3D space as a point mass with a linear momentum. The pedestrian has vector properties such as velocity and acceleration, and other properties such as position, mass, and limitations such as maximum force and speed. The pedestrian is guided by steering forces, which are guided by some basic principles such as target seeking, braking, and collision detection. If an impending collision is detected, the pedestrian slows down and changes its steering forces in an attempt to navigate around the obstacle. An important milestone in this research was the emergence in the simulation of known pedestrian behaviors, such as lane formation, formation of stripes at intersections, and a relationship between speed and density. Current research seeks to include more pedestrian behaviors to improve the model.

Lee et al. (2008) used video surveillance of a hallway to observe the behaviors of pedestrians and the effects of crowding on pedestrians. Various types of data about the pedestrians were recorded, such as walking speed, density, flow rate, trajectory change,
and individual spacing. The study found that walking speed slowed with higher trajectory changes and denser spacing, both in a roughly linear relationship. Demographics of the pedestrians were also recorded, and walking speeds varied with age and gender. Further research in this area includes creating better pedestrian tracking through a robust object detection algorithm, increasing the reliability of the coordinate system used, creating more accurate profiles for the speed and acceleration of each pedestrian, and learning more about behaviors in high-density crowds.

Musse et al. (2004) talk about many different types of crowd modeling, especially simulations. Previously employed techniques include the following: simulations that do not distinguish individuals such as flow and network models, simulations that represent individuals controlled by complex rules based on physical laws, chaos equations, behavioral models, and sociological simulations. There have been two main purposes for crowd modeling: the first purpose has been for simulating particular situations, such as evacuations, and the second purpose has been for high quality visualization for movies or computer games, where realistic behaviors are not a high priority. For simulating evacuations, the main questions being asked of the simulation are: “can the area be evacuated within a prescribed time?” “where do the hold-ups in the flow of people occur?” and “where are the likely areas for a crowd surge to produce unacceptable crushing pressure?” The main challenges in simulating crowds during an evacuation include “a realistic method of collision avoidance, a strong connection with the environment, and a compromise with the numerical and statistical results.”

One commercial product for evacuation simulation is Simulex. Musse et al. (2004) state that “Simulex is computer model aimed at simulation of the escape movement of occupants through large, geometrically complex building spaces defined by 2D floor plans, and connecting staircases. Each individual has attributes such as position,
angle of orientation and walking speed. Various algorithms such as distance mapping, way finding, overtaking, route deviation, and adjustment of individual speeds due to proximity of crowd members are used to facilitate the progress of the simulation.”

Another commercial product in evacuation simulation is Legion. Musse et al. (2004) indicate that in Legion, “every person in the crowd is treated as a virtual individual, with authentic pedestrian attributes drawn probabilistically from empirically established profiles. Each individual scans its local environment and chooses an action that aims to minimize the effort required to arrive at its destination. The decision is made according to the individual’s preferences, location, objectives, and recent experience; and is sensitive to local conditions, context, and intentions of neighbors.”

Besides evacuation, another situation frequently simulated is mass gatherings of people for demonstrations and similar events. Police and military organizations use these types of simulations. A commercial product called CACTUS “is a system developed to assist in planning and training for public order incidents such as large demonstrations and marches. The software designs are based on a world model in which crowd groups and police units are placed on a digitized map and have probabilistic rules for their interactive behavior. The simulation model represents small groups of people as discrete objects” (Musse et al. (2004)).

Shin et al. (2008) describe a model for simulating building evacuations using the software Simulex. The model is used to create evacuation plans especially for multi-purpose buildings, such as buildings with shops and movie theaters, which may be at greater risk for casualties and property damage during emergencies than single-purpose buildings. The calculation of time to evacuate for each person simulated includes time to detect the emergency, time to activate the alarm after detection, time for preparing to evacuate, and movement time. For the movement simulation, a constant movement
speed is randomly assigned to each person, and that speed is reduced if a person moves up or down stairs. Furthermore, the speed is reduced based on a function of occupant density, because people move slower in denser crowds. The effect of door size on occupant density is also taken into account. The model is used to determine the time to evacuate the building based on the shortest exit route method, and a new method for evacuation based on occupant density and exit door capacity is proposed but not tested.

Ko (2003) compares two simulation models for fire evacuation from a building. The first model, Simulex, is a commercially available model that used the “ball-bearing” approach to simulate evacuations. The second model, EvacuatioNZ, is a model developed at the University of Canterbury, which uses more human behaviors in its simulation, especially on the subject of which exit route evacuees would take. EvacuatioNZ’s model resulted in simulations closer to real evacuations in fire drills, which usually took longer to complete than time estimates provided by other models.

Lee and Son (2008) propose a belief-desire-intention (BDI) framework to model the behavior of humans. Specifically, an agent-based model is made for three different evacuation scenarios: a bomb explosion, a bomb threat, and a bomb threat after an explosion. The BDI framework attempts to model a human’s reasoning process by splitting it into three components: beliefs, desires, and intentions. In order to quantify these three components, the BDI makes use of several pre-existing techniques and models, such as extended decision field theory (EDFT), Bayesian belief network (BBN), and Soar (a path-finding algorithm). These techniques make use of the information available to each agent (i.e. simulated person), such as hearing an explosion, seeing smoke, or being informed of an explosion by other agents. In addition, each agent is characterized by its familiarity with the area, risk taking behavior, confidence index (which affects its movement speed), and guidance by police. The results show that a
greater number of agents leads to faster evacuation time since there is a faster exchange of information about the explosion but the effect of crowding is not taken into account.

Hanisch et al. (2003) discusses the use of simulation to model pedestrian flows in large buildings such as airports or train stations. Five types of simulation models are considered. The macroscopic modeling approach models pedestrians with differential equations in a manner that is similar to the modeling of fluids. The entity-based microscopic modeling approach models pedestrians as individual entities that follow a queuing system with a random component. The Cellular Automata based microscopic modeling approach uses a uniform grid of cells to represent the area under study. In Cellular Automata, the state of each cell is affected by its previous state and the state of the surrounding cells, a behavior that corresponds to pedestrian behavior. Multi-agent based microscopic modeling approach defines everything as individual objects, including pedestrians, i.e. agents, and objects in the environment. The environment and the status of neighboring agents affect the behavior of agents.

The mesoscopic pedestrian flow model is the one used by the authors for their simulation. This type of simulation model tracks groups of pedestrians instead of individual pedestrians, and every group has a set of rules governing its behavior. This type of simulation works for the research in the paper because the desired information was the number of pedestrians in an area during a certain time interval, not the individual state of each pedestrian. The rest of the research involved real-time simulation that made short-term predictions of future problems in the pedestrian flows.

Wilson et al. (2006) present a 2D simulation model, called Security Checkpoint Optimizer (SCO), which simulates the inside of an airport and its security checkpoints. The simulation was written in Java making using of open-source packages, and is displayed graphically in a Java Applet that can be accessed via the internet. In this
model, people are simulated as they walk to different places in the airport, and a route-finding portion of the software means the user does not need to create a defined path across a room such as a lobby for people to find their way across. Furthermore, the model simulates other behaviors, such as the queuing of people in line at a security checkpoint. The model was developed for the Transportation and Security Administration (TSA) to evaluate floor plans with measures such as security effectiveness, operational costs, and passenger throughput.

As indicated in the previous two sections, geographic information systems, optimization, and simulation have significant capabilities and when combined their functionality can be significantly enhanced. The annotated bibliography in Appendix A provides additional references of applications of these techniques. The next section describes a more detailed application of geographic information used within an optimization context for hazard zone determination.

4. HAZARD ZONE DETERMINATION VIA OPTIMIZATION

An important aspect of evacuation planning is defining the worst-case evacuation scenario. This allows for an assessment of the risk and vulnerability of a given population area. Through the assessment of risk and vulnerability of an area, emergency planners can formulate better evacuation plans prior to an actual hazardous event. This will improve the chances for a more successful evacuation of people from the hazard area. If emergency planners can be given better tools to prepare for the occurrence of a hazard, the greater the chances of survival of people affected by the hazardous event.

In August of 2005, Hurricane Katrina hit the United States coast and created an estimated $81.2 billion of damage with at least 1,836 people losing their lives. In December of 2004, a tsunami hit Indonesia that caused an estimated death toll of 230,000. These two natural disasters greatly exposed the need for improved emergency
management. With respect to the state of Arkansas, decision support systems for emergency evacuation planning could be potentially utilized if an earthquake was to occur within the New Madrid Seismic Zone, which is located in the Southern and Midwestern United States, including a large portion of the northeastern corner of Arkansas. Even though the last great earthquake (magnitude 7.9) that occurred from this region was in 1812, there is a definite need for closer examination of this area. For these reasons, there are many potential benefits if hazard zone determination is improved.

This section investigates the risk and vulnerability of a defined region building upon the research conducted by Cova and Church (1997) in the hopes of laying a foundation for future research in this area for the long-term goal of having better DSS for emergency planners. However, the immediate contribution to the research field is a comparative analysis of a genetic algorithm to the Cova-Church heuristic for efficiency and effectiveness in defining a worst-case evacuation scenario with corresponding vulnerabilities of areas for a defined region.

The next section examines the relevant literature in the emergency planning area and then focuses on the research of Cova and Church (1997) with a close examination of the overall problem, metrics, and model that was developed in their research. Next, a thorough presentation of Cova and Church’s heuristic is presented with a simple example of the heuristic worked through, to provide a basic understanding of the problem domain. Then an alternative genetic algorithm is presented that may provide for a better determination of hazard zones. Finally, the section wraps up with an evaluation of the two competing algorithms and an application to a real dataset.
4.1. Background and literature on the problem

In the area of emergency planning, there has been a relatively large amount of research that has been conducted that has incorporated GIS with different decision support systems; however, very little has been conducted with respect to risk and vulnerability assessment. The two main areas in which GIS have been incorporated are with simulation and various optimization models that typically concern transportation logistics or resource allocation. These two main types of integration have, however, lacked in the risk and vulnerability assessment. Therefore, for the purpose of this research the focus will be on the improvement of a paper published by Cova and Church (1997) that dealt with the development of a heuristic to aid in the risk and vulnerability assessment of hazard zones or areas affected by some disaster.

In Cova and Church (1997), an integer non-linear optimization model was first developed to assess the transportation difficulties that neighborhoods may face in case of an emergency evacuation. Due to the very low tractability of the model and impracticality of finding guaranteed optimal solutions for the optimization model, they developed a heuristic designed to produce relatively good solutions (not necessarily optimal) for the model.

Their research emphasizes the shift from a temporal to a spatial perspective, which looks at the classification of regions based upon transportation difficulties that might arise during evacuation as opposed to the time it may take to clear a particular hazard zone. Therefore, from this focus, the evacuation difficulty was then required to be measured for each point in the potential hazard zone by finding a zone that contains the point, is limited in size so that evacuation out of the zone can occur, and would represent the difficulty of evacuation for that point for each of the points in the defined region.
The evacuation process in the case of an emergency can be viewed as the “human process that occurs at the level of the individual, and it is assumed that there are a finite number of individuals in any defined region at any given point in time”(Cova and Church, 1997). From this observation, one can then determine that the extent of an evacuation can then be defined as the extent of the population that is involved in the evacuation process. With this perspective and the concept of evacuations at a fixed time, the cardinality of the set of possible evacuation scenarios for a given problem is given in equation (1) from Cova and Church (1997), where \( n \) is the population of the particular study area and \( E \) is the set of all evacuations. From this, one can see the immensity of the number of evacuations as the population grows with a population of only 50 having \( 10^{15} \) potential evacuations at any given point. (Cova and Church, 1997)

\[
|E| = \sum_{i=1}^{n} \binom{n}{i}
\]  

(1)

The first assumption that Cova and Church used is the assumption of contiguity of the defined hazard region. Cova and Church (1997) state that the situation “to which evacuees are not in the proximity is highly unlikely,” and that “contiguity can be added to the definition of a valid evacuation, so that “the population must come from a contiguous area.” The second assumption addressed is the size limit for the constructed model. The most crucial population to evacuate will obviously be the population that is in immediate proximity to the origin of the event; therefore, to focus on the micro-evacuations in this case, a size limit can be enacted. The third assumption addressed the fact that it would be impossible to geo-reference all individuals within a study area”; therefore, “a common aggregate geographic representation of the population must be employed (Cova and Church, 1997)."
For the purpose of this study and the Cova-Church study, the spatial data model was represented as a planar network, where the arcs represented road segments and the nodes represented street intersections. Within this type of representation, which is shown in Figure 1, the darkened areas represent the nodes within a hazard zone with the directed arcs representing an exit from the hazard zone to nodes outside the hazard zone. For this particular example, an evacuee would be able to choose any of the four exit arcs, which in graph theory, would be referred to as a cut-set.

![Figure 1: Graphical Representation of Hazard Zone with Exit Evacuations](image)

The next step in the formulation of the problem is to define a measure for evacuation difficulty, which can be thought of as the “relative effort required to clear an area of its population (Cova and Church, 1997).” Thus Cova and Church (1997) defined a simplistic measure of evacuation difficulty \( ED \), which is shown in equation (2), to be the population \( P \) involved in the evacuation divided by the capacity of the exit choice \( C \).

\[
ED = \frac{P}{C}
\]  

Once evacuation difficulty had been defined, Cova and Church (1997) constructed a spatial classification method using the basic approach of finding a worst-case scenario, which in this case will have the maximum evacuation difficulty, for each
of the nodes in the area of interest. A very important element in defining a location’s spatial evacuation vulnerability is the method in which the evacuation size is limited. When exploring the local neighborhood of each node in the hazard zone, one must first define what is meant by “local.” The five basic ways to define the term “local” are the following: Euclidean distance, network distance, population, area, and node count. Cova and Church (1997) used node count as the size limit such that a maximum number of nodes would be allowed in the hazard zone. Cova and Church (1997) assumed “that network connectivity relations are the most critical component in defining evacuation vulnerability.” In addition, they point out that this is particularly appropriate “in areas where very long dead-end roads extend from urban areas typically to rural areas.” Cova and Church (1997) based their analysis of evacuation vulnerability on an optimization model involving the connectivity of neighborhoods, which were represented by nodes, and the capacity of the connections or arcs between the neighborhoods or nodes. The following section describes the optimization that they developed to capture this concept.

4.2. Formulation of Optimization Model

The following is based on the presentation in Cova and Church (1997). Given a graph $G = (N, A)$ with costs $c_{ij}$ (number of lanes or arcs) and population $a_i$ for each node, one partitions $N$ into two sets $N_1$ and $N_2$ such that $N = N_1 \cup N_2$, $N_1 \cap N_2 = \emptyset$, where $N_1$ is a contiguous partition less than or equal to $s$ (size in nodes) that will contain the root node $i^*$ in order to maximize the weight (population) relative to cost for the arcs between $N_1$ and $N_2$. (Cova and Church, 1997)
Objective function: Maximize \( \frac{\sum a_i x_i}{\sum c_{ij} y_{ij}} \) \hspace{1cm} (3)

Subject to:

\( x_i - x_j \leq y_{ij} \quad \forall i, j \text{ and } \forall j, i \in N \) \hspace{1cm} (4)

\( \sum_i x_i \leq s \) \hspace{1cm} (5)

\( x_{i^*} = 1 \) \hspace{1cm} (6)

\( x_i \in (0,1) \quad \forall i, j \in N \) \hspace{1cm} (7)

\( y_{ij} \in (0,1) \quad \forall i, j \in N \) \hspace{1cm} (8)

Where:

\( x_i = \begin{cases} 1 & \text{if node } i \text{ is in } N_1 \\ 0 & \text{otherwise} \end{cases} \)

\( y_{ij} = \begin{cases} 1 & \text{if node } i \text{ is in } N_1 \text{ and } j \text{ is in } N_2 \\ 0 & \text{otherwise} \end{cases} \)

\( a_i = \text{population weight of node } i \)

\( c_{ij} = \text{cost of arc } ij \)

\( s = \text{maximum size (in nodes) of } N_1 \)

\( i^* = \text{index of root node in } N_1 \)

In this formulation, the objective function (3) maximizes the relative evacuation difficulty, which is measured as the weight or population involved divided by the total cost of the arcs between the two sets \( N_1 \) and \( N_2 \). Constraint (4) ensures that if node \( i \) is in the partition and node \( j \) is not, then the arc between the two \( y_{ij} \) must be equal to 1 or in other words be included in the exit set. Furthermore, if an arc, \( y_{ij} \), is included in the exit set then the cost for that arc is included in the objective function. Constraint (5) limits the
search partitions to where $N_1$ is less than the size limit ($s$). Constraint (6) states that the root node $i^*$ must be included in the set $N_1$. Constraint (7) and (8) ensures that $y_{ij}$ and $x_i$ are binary integer variables. Once a solution has been reached, that solution is then checked for contiguity and if the solution is not contiguous then a constraint is then developed to eliminate this as possible solution and the optimization is resolved until the optimal solution reached is that of a contiguous hazard zone. (Cova and Church, 1997)

To aid in the understanding of this optimization model, a simple example was constructed to show how each of the constraints and objective function value can be calculated. As indicated in the graphical representation of the problem in Figure 2, there are four nodes in the example with node 1 being the only node in the hazard zone and the connectivity as shown.

![Figure 2: Graphical Representation for Example Problem](image)

Assume that the population at each of the nodes is 30, the capacities of one between node 1 and node 2, node 1 and node 3, and node 1 and node 4. This can be seen in the following equations. Furthermore, a maximum hazard zone size ($s$) of two was chosen. All variable values not specified can be assumed to be 0.

\[
\begin{align*}
& a_1 = a_2 = a_3 = a_4 = 30 \\
& c_{12} = c_{21} = c_{13} = c_{31} = c_{14} = c_{41} = 1 \\
& y_{12} = y_{13} = y_{14} = 1 \\
& x_i = 1
\end{align*}
\]

Next, constraint (4) is evaluated for every combination of $i$ and $j$; however, only a subset of those are shown to illustrate the calculation method.
Next, constraint (5) was evaluated and determined to be less than the specified \( s \).

\[
x_1 = x_1 = 1 - 1 = 1 
\leq y_{11} = 0 
\]
\[
x_1 = x_2 = 1 - 0 = 1 \leq y_{12} = 1 
\]
\[
x_1 = x_3 = 1 - 0 = 1 \leq y_{13} = 1 
\]
\[
x_1 = x_4 = 1 - 0 = 1 \leq y_{14} = 1 
\]

Constraints (6), (7), and (8) can be visually verified due to the fact that the root node (node 1) is equal to 1, and all \( x_i \) and \( y_{ij} \) values are either 0 or 1. Therefore, the objective function value or evacuation difficulty can be calculated.

\[
ED = \frac{a_1 x_1}{c_{12} y_{12} + c_{13} y_{13} + c_{14} y_{14}} = \frac{30}{1 + 1 + 1} = 10 
\]

The following section presents the heuristic procedure developed by Cova and Church for solving the non-linear optimization model.

4.2.1. The Cova-Church Heuristic

Due to the very low tractability and difficulty finding a guaranteed optimal solution for practical applications with the nonlinear integer optimization model, Cova and Church (1997) developed a heuristic to find solutions in an efficient manner. The binary integer constraints on the decision variables cause the problem to be classified as nondeterministic polynomial-time hard or NP-hard with \( 2^k \) potential solutions, where \( k \) is the number of decision variables. For example, a problem with 200 binary variables would have more than \( 10^{60} \) potential solutions. Therefore, algorithms and heuristics come into consideration to get relatively good solutions in much less time than solving the nonlinear integer optimization model. Furthermore, the optimization model can result in a noncontiguous hazard zone and the only way to eliminate this is to then construct a constraint that will eliminate this as a possible solution and then resolve the model.
Heuristics are typically developed through experience-based knowledge of a particular problem design. Typically when constructing a model to represent a real-world problem, there are some assumptions that may be questionable; therefore, with respect to recognizing that there may be issues with the assumptions of any model, it can be understood that it may be better to achieve a reasonable or non-optimal solution to a more accurate model than to attempt to find the optimal solution to an inaccurate or oversimplified model of the real world as Silver (2004) articulates. Thus, heuristics are typically utilized for the five following reasons as described by Silver (2004):

- Facilitate the implementation of the solution (easier to live with a problem than to implement a solution that cannot be understood)
- Improvement over current practices (if the heuristics improves the current situation)
- Reasonably fast results
- Robustness (less sensitivity to variation)
- In optimization such as eliminating solution space for faster optimization

With this basic heuristic information, the next section will discuss the general methodology or constructions of heuristics.

In constructing the cluster of nodes or group of nodes that will be considered in the hazard zone given the root node, the adjacent nodes are evaluated by determining each node's potential gain in the objective function value given by equation (9). In equation (9), \( C_k \) is the total exit capacity out of the cluster or hazard zone to nodes outside of this cluster, \( P_k \) is the total population that is affected by the hazard or total population in the hazard zone, \( a_i \) is the population of the node that is a potential candidate to be added to the hazard zone, \( o_i \) is the exit capacity that would be open for use out of the hazard zone if node \( i \) is included in the hazard zone, and \( o_c \) is the exit capacity that would be closed by adding node \( i \) due to the fact that if node \( i \) is added then the capacity from
the hazard zone to node $i$ will no longer be exiting the hazard zone. Therefore, one can see that from the objective function (the affected population ($P$) divided by the exit capacity ($C$)), one can derive $g_i$ or the potential gain in the objective function from adding potential candidate node $i$.

$$\frac{P_{k+1}}{C_{k+1}} = \frac{P_k + a_i}{C_k + (a_i - q_c)} = g_i \frac{P_k}{C_k} \quad \text{or} \quad g_i = \frac{C_k(P_k + a_i)}{P_k(C_k + (a_i - q_c))} \quad (9)$$

where:

$k =$ index of iteration

$g_i =$ gain in the objective function if node $i$ is selected

$P_k =$ total population of cluster at iteration $k$

$C_k =$ total exit capacity of cluster at iteration $k$

$a_i =$ population of node $i$

$a_i =$ new capacity node $i$ would open, if selected

$q_c =$ existing exit capacity node $i$ would close, if selected

The heuristic starts with the root node $i^*$ in set $N_1$ then looks at adding adjacent nodes to the set. If these adjacent nodes have a potential gain in the objective value ($g_i$) that is greater than the defined alpha value, the node is then put in a candidate list for possible selection. Once all adjacent nodes have been looked at, a node to add to set $N_1$ is randomly chosen from the candidate list. This process is then repeated until the set $N_1$ has reached the size limit ($s$) or until there are no candidate nodes. A top-level explanation of this algorithm can be seen in Exhibit 1. A more in depth explanation of the interpretation of this algorithm for the purpose of this research will be discussed in the section concerning modeling issues with the following section walking through a simple example using the algorithm for better understanding of how the algorithm works.
4.2.2. Example Operation of the Cova-Church Heuristic

For a better understanding of how the algorithm functions, a simple example with 10 nodes was constructed. The ten nodes are connected to one another by arcs, which represent the potential capacity from one node to another. The nodes that are connected to one another are considered adjacent nodes. For the purpose of the study, the parameters utilized in the calculations are defined as follows:

\[
a_i = \text{aggregate population at node } i, \text{ which is representative of the total population within each of the defined regions}
\]

\[
t = \text{number of starts or times the algorithm will start with each of the ten nodes}
\]

\[
\alpha = \text{cutoff point for potential gain in objective function}
\]

\[
s = \text{maximum hazard zone size}
\]

The number of starts \((t)\) was set to one. Due to randomization in the selection of a node added to the hazard zone from the candidate list of nodes, it could be possible that different starts may result in different final evacuation difficulties; therefore, multiple starts would typically be required for problems. The alpha level \((\alpha)\) was then set to 0.90;
therefore, nodes that have a potential gain when added to the hazard zone less than the predefined alpha level will not be considered to be potentially added to the hazard zone. Furthermore, the maximum size of the hazard zone (s) was set to five nodes; therefore, a maximum of five of the ten nodes can be in the final hazard zone. Each step of the algorithm with one start and beginning with root Node 1 is depicted in the following example with the corresponding final worst case scenario solution. The final worst case evacuation scenario results beginning with each of the remaining nodes was then calculated in the same manner and the results of each were given. The example problem is illustrated in Figure 3 with the number assignment of each of the nodes. Furthermore, Table 1 shows the aggregated populations that were assigned to each of the ten nodes for the example problem.

![Figure 3: Example Population Graph with Nodes and Arcs](image_url)
Table 1: Aggregated Populations at Each of the Ten Nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

Start at Node 1, Iteration 1

For the first iteration, the root node 1 was given to be in the hazard zone, which is indicated by the lighter gray shaded region in Figure 4 and the two potential candidate nodes to be added into the hazard zone are nodes 2 and 4, which is indicated by a darker gray shaded region in Figure 4. The first step is to calculate the current hazard zone’s evacuation difficulty (ED), which is the objective function for the heuristic. Next, the evacuation difficulty was calculated along with the potential gain value ($g_i$) for each of the candidate or adjacent nodes $i$.  

Figure 4: Heuristic Example Problem Iteration 1
Nodes 2 and 4 are then put into the candidate list because both have potential gains greater than $\alpha = 0.90$. Node 4 was then randomly selected to be included in hazard set ($N_1$) and now node 1 and 4 are in the hazard zone.

*Start at Node 1, Iteration 2*

For Iteration 2, Node 1 and 4 are in the hazard zone, which is indicated by the lighter gray shaded area in Figure 5 and Nodes 2, 3, 5, and 8 are the potential candidate nodes that are indicated by the darker gray shaded area in Figure 5. The corresponding evacuation difficulty (ED) was calculated along with the potential gain value ($g_i$) for each of the candidate nodes $i$.

\[
ED = \frac{P}{C} = \frac{19}{2} = 9.5
\]

\[
g_2 = \frac{P_1(a_1 + a_2)}{P_1(a_1 + a_2)} = \frac{2(19 + 27)}{19(2 + 1 - 1)} = 2.42
\]

\[
g_4 = \frac{P_1(a_1 + a_4)}{P_1(a_1 + a_2)} = \frac{2(19 + 34)}{19(2 + 3 - 1)} = 1.3947
\]

![Figure 5: Heuristic Example Problem Iteration 2](image)
Nodes 2, 3, 5, and 8 are put in a candidate list because they have potential gains greater than $\alpha$. Node 8 was then randomly selected to be included in hazard set ($N_1$) and now node 1, 4, and 8 are in the hazard zone.

*Start at Node 1, Iteration 3*

For iteration 3, node 1, 4 and 8 are in the hazard zone, which is indicated by the lighter gray shaded area in Figure 6 and nodes 2, 3, and 5 are the potential candidate nodes that are indicated by the darker gray shaded area in Figure 6. The corresponding evacuation difficulty (ED) was calculated along with the potential gain value ($g_i$) for each of the candidate nodes $i$.

![Figure 6: Heuristic Example Problem Iteration 3](image-url)
\[ ED = \frac{66}{3} = 22 \]
\[ g_2 = \frac{3(66 + 27)}{66(3 + 1 - 1)} = 1.0910 \]
\[ g_3 = \frac{3(66 + 6)}{66(3 + 1 - 1)} = 1.0910 \]
\[ g_5 = \frac{3(66 + 36)}{66(3 + 3 - 1)} = 0.9273 \]

Nodes 2, 3, and 5 are put in a candidate list because all have potential gains greater than \( \alpha \). Node 2 was then randomly selected to be included in hazard set \( (N_1) \) and now node 1, 2, 4, and 8 are in the hazard zone.

*Start at node 1, Iteration 4*

For Iteration 4, Node 1, 2, 4 and 8 are in the hazard zone, which is indicated by the lighter gray shaded area in and Nodes 3, 5, 7 and 9 are the potential candidate nodes that are indicated by the darker gray shaded area in Figure 7. The corresponding evacuation difficulty (ED) was calculated along with the potential gain value \( (g_i) \) for each of the candidate nodes \( i \).

![Figure 7: Heuristic Example Problem Iteration 4](image-url)
Nodes 3, and 5 are put in a candidate list because both have potential gains greater than \( \alpha \); whereas, node 7 and 9 were eliminated because it had a potential gain less than \( \alpha \). Node 5 was then randomly selected to be included in hazard set \((N_1)\) and now nodes 1, 2, 4, 5, and 8 are in the hazard zone.

The final hazard zone set \((N_1)\) includes nodes 2, 5, 6, 9, and 10 with an evacuation difficulty \((ED)\) of 43 as shown in Figure 8.

\[
ED = \frac{93}{3} = 31
\]

\[
g_3 = \frac{3(93+6)}{93(3+1-1)} = 1.0645
\]

\[
g_5 = \frac{3(93+36)}{93(3+2-2)} = 1.3871
\]

\[
g_7 = \frac{3(93+18)}{93(3+2-2)} = 0.8952
\]

\[
g_9 = \frac{3(93+20)}{93(3+2-0)} = 0.7290
\]

After one start at each of the ten nodes, the following hazard zone with the corresponding evacuation difficulty was calculated and is shown in Table 2. From this result, the worst case hazard zone was determined to include nodes 2, 5, 6, 9, and 10 with an evacuation difficulty of 58.5 people per exiting road.
Table 2: Final Worst Case Evacuation Difficulty for Each Root Node

<table>
<thead>
<tr>
<th>Starting Node</th>
<th>Hazard Zone</th>
<th>Evacuation Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,2,4,5,8</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>1,2</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>1,2,3,4,5</td>
<td>30.5</td>
</tr>
<tr>
<td>4</td>
<td>1,3,4,7,8</td>
<td>45</td>
</tr>
<tr>
<td>5</td>
<td>2,5,6,9,10</td>
<td>58.5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>3,7</td>
<td>24</td>
</tr>
<tr>
<td>8</td>
<td>1,3,4,7,8</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>4,5,6,8,9</td>
<td>31.25</td>
</tr>
<tr>
<td>10</td>
<td>2,5,6,9,10</td>
<td>58.5</td>
</tr>
</tbody>
</table>

From the results of this simple example, the worse case evacuation scenario would be a hazard zone that includes nodes 2, 5, 6, 9, and 10 with an evacuation difficulty of 58.5, which can be interpreted as an average population of 58.5 being evacuated through each exit capacity. Furthermore, the heuristic with only one start, an alpha level ($\alpha$) of 0.90, and maximum hazard zone of 5 had two different root nodes (nodes 5 and 10) result in the worst case evacuation scenario. This solution was also verified to be the optimal evacuation difficulty with a simple total enumeration code that was developed for this example problem. Through solving the example problem, there were a couple of major critiques of the assumptions made by Cova and Church (1997) in the construction of this heuristic, which will be discussed the following sections.

4.2.3. Genetic Algorithm

For the purpose of this research, a genetic algorithm can be broadly defined as a search technique that is used to find exact or approximate solutions to an optimization problem.
Genetic algorithms can be categorized as global search heuristics that are in a class of evolutionary algorithms which are inspired by biological forces and contain properties such as immigration, mutation, selection, and crossover. The basic structure of a genetic algorithm consists of an initial population generation that includes chromosomes, or potential solution vectors. Once the initial population has been generated, the fitness of each chromosome is then determined. Once the fittest chromosome can be distinguished from less fit chromosomes, a user-specified percentage of the fittest chromosomes are then selected. Next, one of two events occurs. In the first option, the chromosome is mutated, which entails the manipulation of some small aspect of the chromosome. In the second option, the chromosome is bred using a technique called crossover, where parts of two chromosomes are used to make one chromosome that could potentially have a better fitness than the parent chromosomes. Immigration, or the addition of a randomly generated chromosome, is also a key part of many genetic algorithms. The fitness of each of the new chromosomes is then determined and the fittest chromosomes of the new population, which includes the new chromosomes, are then selected. This process is then repeated until the termination criteria specified by the user is reached. The typical pseudo-code for a genetic algorithm can be seen in Exhibit 2.

<table>
<thead>
<tr>
<th>Step 1:</th>
<th>Generate the initial population.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2:</td>
<td>Evaluate the fitness of each individual in the population.</td>
</tr>
<tr>
<td>Step 3:</td>
<td>Select the fittest individuals for reproduction.</td>
</tr>
<tr>
<td>Step 4:</td>
<td>Breed the new generation through crossover, mutation, and/or immigration.</td>
</tr>
<tr>
<td>Step 5:</td>
<td>Evaluate the individual fitnesses of the offspring.</td>
</tr>
<tr>
<td>Step 6:</td>
<td>Replace the least fit part of current population with the offspring which is more fit.</td>
</tr>
<tr>
<td>Step 7:</td>
<td>Repeat Steps 3 - 6 until the termination criteria has been met.</td>
</tr>
</tbody>
</table>

Exhibit 2: General Pseudo-code for a Genetic Algorithm
4.3. Testing and Evaluation Methodology

For the purpose of addressing the modeling issues associated with this research, it is noted that all coding was conducted in Microsoft Visual Basic and all experiments were conducted on computers with identical specifications. The following sections address modeling issues required during testing and evaluation. The modeling issues were: generation of problem instances, algorithm implementation, and methods used for the evaluation.

4.3.1. Problem Instance Generation

The first modeling issue is how to generate the problem instances which are used as the basis for comparing the genetic algorithm and the Cova heuristic. According to the definition of the mathematical program developed by Cova and Church, the basic problem instance can be structured as a graph of aggregated populations at nodes and arcs, each with predefined capacities. For the purpose of this research, a problem instance will be represented in this manner by using a matrix. The capacities from one node to another is set to either a 1, which would show that there is a travel capacity between nodes (arc present between nodes), or a 0, which indicates that there is no travel capacity (no arc between nodes). Furthermore, it will be assumed from this point forward that when a problem instance is abstracted from a realistic application to a graphical representation, the road network from one location to another will be grouped into one defined capacity from one location to another. For example, instead of allowing three arcs, each with a defined capacity, to connect one node to another, only one arc, whose capacity is the sum of the three arcs, will be represented in the structure.

For the purpose of generating problem instances, which will be utilized in examining the genetic algorithm, basic random generation algorithms designed for graphs
may be used. The first key element that will be randomly generated is the number of
nodes \((n)\) in the problem instance. For experimental purposes, four types of problem
instances were generated. First, a small set of problem instances with 20 nodes was
generated so that total enumeration could be utilized in solving these problem instances
due to the fact that problem instances with more nodes required an immense amount of
time. Second, a set of problem instances with 32, 64, and 128 nodes was generated so that
the genetic algorithm and Cova heuristic could be used to solve problem instances with
absolute minimum \((n-1)\) to absolute maximum \((n\times(n-1))/2\) number of connections or
arcs. The third type of problem instance is based on data that mimics real world networks
with a minimum of \(n\)-1 arcs and maximum of \(4n\). The fourth and final problem instance
type was a real world scenario abstracted from a real world network by John H. Wilson in
the Center for Advanced Spatial Technologies at the University of Arkansas.

The process of generating a random connected graph is itself a topic of much
research; therefore, for this project, code \((A \text{ Graph Generation Package})\) that was
developed by Dr. Richard Johnsonbaugh and Dr. Martin Kalin of the Department of
Computer Science and Information Systems at Depaul University was utilized for the
generation of random connected graphs. See reference [56]. This code was slightly
modified to generate a specified number of replications for three different levels of
connectivity densities for the purpose of better comparing the Cova heuristic and the
genetic algorithm.

For each of the node levels of the first and second types of problem instances, the
minimum number of arcs that can be present for the graph to still be fully connected is \(n-1\),
while the maximum is \((n-1)\times n)/2\). This is a range of \((31, 496)\) for the 32 node
problem instances, \((63, 2016)\) for the 64 node problem instances, and \((127, 8128)\) for the
128 node problem instances. The range was then divided into three equal levels of
connectivity densities: low, medium, and high. The code was then used to generate random connected graphs with the three levels of nodes and the three levels of connectivity densities. Once the Johnsonbaugh code had generated the graphs and the connectivity (arc set) information, the number of nodes was then input into the random input generation code which randomly generated three levels of population density (low, medium, and high) and two levels of hazard zone size (small and large).

For analysis purposes, the input file generation code generated three population densities and two hazard zone sizes that were then assigned to each connection density for all combinations. In other words, a problem instance with a low population and a large hazard zone size would have the same assigned population values as a problem instance with a low population and a small hazard zone size, so that when hazard zone sizes are compared, all other aspects are held constant with only the hazard zone size being varied.

From the Cova-Church paper, it was mentioned that a high population density would be 500 people for the level of analysis that they conducted; therefore, for this research, a small population was considered to between one person and 250 people and a large population was considered to be between 251 and 500 people. From this assumption for what is considered to be a large and small population for this experiment, each node was then randomly classified as either having a small population or a large population according to the discrete distribution for the population density specifications and a random population was generated accordingly for each node and connection density combination.

For the population density levels, the low population density level had 75% of its nodes with a small population and 25% with a large population, the medium population density level had 50% of its nodes with a small population and 50% with a large
population, and the high population density level had 25% of its nodes with a small population and 75% with a large population. Next, the two levels of hazard zone size (small and large) were then randomly generated for each of the preceding problem instances. The minimum hazard zone size possible is one node and the maximum hazard zone size possible is \( n-1 \), because all nodes cannot be in the hazard or else no solution could result. The range of \((1, n-1)\) was then divided in half to get the small and large hazard zone ranges for each of the node levels.

For the third problem instance type, the problem instances were generated in a similar manner as the first and second problem instance type except that the connection density was limited to the range of \( n-1 \) to \( 4n \) to better mimic the connectivity found within road networks. The maximum arc limit of \( 4n \) derives from the most common maximum intersection of roads (4-way intersection) in a real-world network. Furthermore, the only difference affecting population density was that the small population nodes had a range of 1 to 10,000 and the large population density nodes had a range of 10,001 to 20,000. The hazard zone levels were consistent with the first and second problem instance type.

Once a problem instance is generated, the problem can then be solved using the Cova and Church (1997) heuristic and the genetic algorithm with a small set being totally enumerated.

4.3.2. Algorithm Implementations

4.3.2.1. Enumeration

For the purpose of comparison between the genetic algorithm and the Cova heuristic for a 20 node set of problem instances, code for the total enumeration of the all potentially possible solutions was developed. The total enumeration code will be used to determine
how “close” the genetic algorithm and the Cova heuristic are to achieving the optimal solution. For the larger node sets (32, 64, and 128), however, total enumeration was not an option because the solution space doubles with every additional node, making the process very time intensive. For instance, problem instances with 64 nodes would take an estimated time of $2.64 \times 10^{13}$ hours to complete each problem instance scenario.

With respect to coding of the total enumeration, the first issue was that of the code being dynamic. The original code that was used to validate the solution to the example problem was hard-coded for the specific example; however, this was not feasible to use for evaluating multiple problems that vary in number of nodes in the problem and/or the maximum hazard zone size. Therefore, a strategy that uses subtraction in binary math was utilized and will be explained in detail.

First, the code takes as inputs the number of nodes in the problem, the maximum size of the hazard zone allowed, the population of each node, and the adjacency matrix values associated with the connectivity of the graph. Arrays are created for each of the inputs of the adjacency values, which will either be a 1 for a node considered “adjacent” to another node or a 0 for a node that is not “adjacent” to another node. The capacity value is input as the capacity from one node to another “adjacent” node. The arrays, the capacity values, and the population that is aggregated at each node are assigned according to the problem instance specifications. One must remember from previous sections that a node is considered “adjacent” if and only if there is a connection, or arc, between the two nodes and may not be necessarily correlated with the spatial adjacency of the nodes. The values are then read into the code for each of the remaining input parameters.

Next, the total enumeration code finds the maximum value in binary for a potential solution. For example, a problem with ten nodes and a maximum hazard zone size of five will have a binary representation of 1111100000, which is 992 in integer
form. For the example problem of ten nodes, the totally enumerated solutions are equal to \(2^{10}\) or 1024; therefore, 3.125\% of the generated decision vector values that are infeasible with respect to the maximum hazard zone size constraint should be eliminated from the evaluation process. This may become more important for larger problems.

Equation 9 illustrates the how this maximum binary representation is found in integer form.

\[
\sum_{j=0}^{n-1} 2^{n-j-1}
\]  

(9)

Once the maximum integer value for the binary form is calculated, the total enumeration code then starts with this value and decrements until reaching the value of 1, which is the smallest value possible given only one node, which is the last node in the node in the list of nodes included in the hazard zone. The integer value is then converted into a string of binary values, which forms a potential solution decision vector. The binary values in the string consist of a 0 for nodes not included in the hazard zone of the potential solution’s hazard zone or a 1 for nodes that are included in the hazard zone of the potential solution. However, it must be noted that strings that have lengths less than the number of nodes specified in the problem are supplemented with additional zeros for the first \((n - \text{length of binary string})\) nodes.

Once the decision vector has been constructed, the constraints of hazard zone size and adjacency, which is an attempt at deriving a contiguous area for the final solution, are then evaluated for the generated decision vector. If the decision vector is within the hazard zone size constraint and all nodes that are in the hazard zone are adjacent to another node in the hazard zone, then the non-linear objective function for evacuation difficulty is evaluated for the that particular decision vector. If this is a better solution to the problem, the respective decision vector, the corresponding objective function value,
the total affected population, and the exit capacity are stored. Once all needed solutions have been generated and evaluated the best solution is then given to the user.

The most crucial issue with the total enumeration code involves the same problem that the nonlinear integer optimization model discussed earlier. There is a possibility that a solution with multiple hazard zone groupings is possible. For example, with the ten node example previously described, the problem could have been constructed in such a way that the optimal solution had two distinctive hazard zones with multiple nodes in each due to the fact that the adjacency constraint has so far only been conceptualized to check that each node is adjacent to another node in the hazard zone. For this issue, a reduced matrix is constructed with only the nodes in the potential solution and the transitive closure matrix computed for the reduced adjacency matrix. Each element of the transitive closure matrix is checked to ensure that each node in the hazard zone of the potential solution is reachable by every other node in the hazard zone of the potential solution. This matrix represents the connectivity relationship of the nodes included in the hazard zone for the potential solution. For example, if nodes $a$, $b$, and $c$ are in the hazard zone of the potential solution and node $a$ is connected to node $b$, then $a$ to $b$ would have a 1 in the corresponding transitive closure matrix element. Furthermore, if nodes $b$ and $c$ are connected, then $b$ to $c$ and $a$ to $b$ would have a 1 in the corresponding transitive closure matrix element. Once the transitive closure matrix has been constructed, then each of the corresponding elements for the nodes in the hazard zone of the potential solution is checked to ensure that each node in the hazard zone of the potential solution is reachable by all other nodes in the hazard zone.
4.3.2.2. Cova Heuristic

The heuristic developed by Cova and Church is designed to directly address the issue of having a contiguous hazard zone that causes difficulty in solving the formulated optimization model. The basic premise of the heuristic is to start with one node and only look at adding adjacent nodes that meet a certain alpha level or a potential gain in the objective function. By structuring the code in this manner, the final solution suggested by the heuristic is forced to be contiguous.

The Cova heuristic inputs the data from the defined problem in the same manner as the total enumeration code. First, the code takes the inputs of the number of nodes in the problem, maximum size of the hazard zone, the alpha level, and the number of starts. Arrays for each of the inputs of the adjacency values, the capacity value, and the population that is aggregated at each node are declared based upon the inputs of the number of nodes that are in the problem instance. The values are then read into the code for each of the remaining input parameters.

Next, the code will set the root, or beginning node, to the value 1, which implies that it is in the hazard zone, and all other nodes are set to the value of 0, which implies they are not in the hazard zone. Every node in the problem instance will systematically become the root node. Once the root node has been chosen, each node that is adjacent to the root node that is not currently in the hazard zone will be evaluated according to the net change in the exit capacity and the added population. This added population will then be used to calculate the potential gain in the objective function that the node causes if added to the hazard zone. The adjacent nodes that have a potential gain greater than the predetermined alpha level will be stored in a list, and one node will be randomly selected to be added to the hazard zone. This process of adding adjacent nodes will be repeated
until either there are no nodes with a potential gain greater than the alpha level or the maximum hazard zone size has been reached.

Once one of these ending criteria has been reached, the objective function value is compared to the current greatest objective function value. The objective function value, the decision vector, and corresponding affected population, and exit capacity are then stored for the greatest objective function value at that point in time. Each node will be set as the root node and the code evaluated for the number of starts defined by the user. The main reason for having multiple starts at each root node is because the adjacent node to be added to the hazard zone is selected randomly from a list of candidates. Once all starts have been completed for each of the nodes in the problem instance, the final solution is output to the user.

For the purpose of coding the heuristic, some further assumptions other than the ones specified before had to be made. First, there was an assumption that was inferred from the Cova and Church paper that adjacency can only come through connectivity. Second, there was an implicit assumption that during the time of evacuation that the capacity from one node to another would be symmetrical; thus, the capacity from one node to a second node would be the same as the capacity from the second node to the original node. The next section discusses the preliminary investigation of a genetic algorithms applied to the problem.

4.3.2.3. Genetic Algorithm

The genetic algorithm constructed for comparison with the Cova heuristic addresses the issues of randomly generated initial populations, crossover, mutation, immigration, population diversity, and reachability for a contiguous hazard zone. All of these issues will be addressed in this section.
First, the number of nodes in the problem instance, the population of each of those nodes, the adjacency matrix values, and maximum hazard zone size are read into the genetic algorithm code from input files with dynamic dimensioning of all arrays with respect to the number required for each of the node levels being tested. Next, the specifications of the genetic algorithm are set. For the purpose of this study, the initial population size will be 100 potential solutions with a diversity percentages of 30% (or 30 chromosomes) being kept from each previous iteration, 40% (or 40 chromosomes) from crossover, 20% (or 20 chromosomes) from mutation, and 10% (or 10 chromosomes) from immigration.

Once the initial random population of 100 chromosomes have been generated, the fitness of each chromosome is then determined through the calculation of the evacuation difficulty for that potential solution in which the total affected population and corresponding exit capacity of the hazard zone of the potential solution chromosome is used. When evaluating a potential solution, if the number of nodes in the hazard zone is greater than one and less than the maximum hazard zone for the problem instance, then the evacuation difficulty is calculated with no penalty. However, if there are more nodes than allowed, a penalty factor is used to adjust the evacuation difficulty in an attempt to drive down the evacuation difficulty of the infeasible potential solutions with respect to how far the potential solution is from being feasible. This penalty factor is calculated by using the maximum population associated with the nodes in the problem instance multiplied by the square of the difference of the hazard zone of the infeasible solution and the maximum allowed multiplied by a user specified penalty value.

Once the fitness of each of the chromosomes has been calculated, the list of the potential solutions is then sorted using a quicksort sorting algorithm that sorts the chromosomes according to their fitness function values. Quicksort takes an average of
\(O(n \log n)\) comparisons and a maximum of \(O(n^2)\) comparisons. However, this algorithm typically runs significantly faster than algorithms of similar time complexity and, thus, was chosen in an attempt to reduce the time spent sorting the chromosomes. This algorithm finds a pivot point, reorders the elements into those above the pivot point and below the pivot, and recursively sorts the sub-lists until all elements are sorted. Once all chromosomes have been sorted in descending order by fitness value, the genetic algorithm starts at the first potential solution, which when sorted would be the fittest, and checks the reachability of each of the chromosomes using the same reachability function as in the total enumeration code until a potential solution with both a feasible hazard zone size and contiguous hazard zone has been found.

This chromosome is then placed in the first element of the keep selection list, thus ensuring that the best feasible solution will always be in the first element of the potential solution array. Next, starting from the first potential solution in the original sorted list, a diverse keep selection population is stored into the keep selection list. The genetic algorithm ensures a diverse keep selection by first comparing a selected chromosome’s evacuation difficulty to the last stored one in the list, which reduces time and works in this case because the potential solutions have been sorted. If they are equal, then each element of the chromosome is compared to see if it is the same solution or just another hazard zone combination with the same evacuation difficulty. If it is a different solution, then that solution is stored in the keep selection list. If it represents the same solution, then the next potential solution is evaluated to see whether it should be added to the keep selection list until the keep selection of 30% has been met. If there are not enough chromosomes to fill this keep selection amount, then a simple mutation is conducted on randomly selected chromosomes until the keep selection specification has been met with
a diverse list of potential solutions as checked with the preceding diversity method. This mutation process will be discussed later in this section.

Once the keep selection of the potential solutions has been determined, the process of one-point crossover, or breeding, is conducted. First, two chromosomes are randomly selected from the keep selection potential solutions. Next, a random point or element within these chosen chromosomes is then selected. Following that, a new chromosome, called the child chromosome, is created by storing the elements of one of the randomly selected chromosomes up to the random point and the elements of the other randomly selected chromosome are stored from the random point to the end of the elements in the chromosome. During this process, the hazard zone of the newly created chromosome is calculated during the storage process of the elements from each of the randomly selected chromosomes. Once the child chromosome has been made, the fitness of this chromosome is then calculated in the same manner as the initial population chromosomes with penalty factors where necessary.

Next, the process of mutation is implemented to created new chromosomes for evaluation. First, a chromosome in the keep selection is randomly selected for mutation and a random element is selected to be mutated. The element in the randomly selected mutation point is then changed and the hazard zone of the potential solution is updated. If the selected element contained a 0, then it will be changed to a 1 and the hazard zone size will be increased by 1. If the selected element contained a 1, then it will be changed to a 0 and the hazard zone size will be decreased by 1. Once the new chromosome has been created, the fitness of this chromosome is then calculated in the same manner as the initial population chromosomes with penalty factors where necessary.

Next, 10% of the new population is created through the immigration process. The immigration process creates new chromosomes in the same manner as the initial
population. This 10% is just randomly generated new chromosomes. This attempts to help keep the genetic algorithm from being stuck in local optima that is not also the global optimum. Once the new chromosomes have been created through the immigration process, the fitness of this chromosome is then calculated in the same manner as the initial population chromosomes with penalty factors where necessary.

Once the population has been restored to 100 from the 30% kept from the previous iteration, crossover, mutation, and immigration, the new population is then sorted using the same recursive quicksort algorithm as used before. Once the population has been sorted, a diverse 30% of the fittest chromosomes are stored in a temporary list. However, one must note that no mutation process will be needed at this point to ensure diversity due to the fact that if the first 30% (or 30 for this experiment) kept are diverse, then the following iteration, as well as every iteration after that, will have at least 30 that are diverse because the initial 30 are diverse.

After each iteration, the evolution of the genetic algorithm is checked to determine whether the termination criteria have been met. For this experiment, the termination criteria of choice is the number of consecutive iterations with a feasible contiguous hazard zone size and less than a 0.1% increase or non-improving evacuation difficulty of the best solution. The early termination criteria implemented was determined through initial test trials of varying levels of consecutive non-improving best solutions for each of the node levels; however, a maximum number of 7000 iterations was set for all experimental trials.

Finally, the number of nodes, connection density, population density, maximum hazard zone size, time to solution, actual hazard zone size of solution, number of iterations before termination, evacuation difficulty of final solution, total affected population of final solution, exit capacity of final solution, and final solution elements
(0’s and 1’s of which elements are in the hazard zone) were stored for analysis. In the subsequent section, the evaluation of the algorithm and the experimental design is discussed.

4.3.3. Algorithm Evaluation and Experimental Design

The experiment for the comparison of the genetic algorithm and the Cova heuristic was designed with four basic types of problem instances. The first type consists of the validation of both the genetic algorithm and Cova heuristic through an initial pilot test trial using the example problem developed for the walkthrough of how the Cova heuristic works. In the second type, experiments were conducted on a random set of 20 node problem instances in which three levels of connection density (low, medium, and high), three levels of population density (low, medium, and high), and two levels of maximum hazard zone size (small and large) were generated as described in the Problem Instance Generation section. The genetic algorithm and the Cova heuristic were then used to find the worst case solution that was then compared in time and closeness to optimality with one another. The optimal solution was determined through the total enumeration code; however, larger node problem instances were unable to be totally enumerated due to the immense amount of time required to totally enumerate these problem instances. For instance, one of the 64 node problem instances would require an estimated $2.64 \times 10^{13}$ hours to totally enumerate.

The second type of problem instances used for analysis consists of 32, 64 and 128 node problem instances in which three levels of connection density (low, medium, and high), three levels of population density (low, medium, and high), and two levels of maximum hazard zone size (small and large) were generated as described in the Problem Instance Generation section. This type of problem instances looked at the entire spectrum

69
of connection density levels from the absolute minimum arcs required to make a connected graph to the absolute maximum so that more than just real world applicable instances could be evaluated. The genetic algorithm and Cova heuristic were then used to solve these problem instances and compared to one another based upon time to find a recommended solution and evacuation difficulty. Total enumeration was unable to be used to determine the optimal solution due to the immense amount of time required.

The third type of problem instances consists of 128 node problem instances that were generated for more real world applications. For this analysis, three levels of connection density (low, medium, and high), three levels of population density (low, medium, and high), and two levels of maximum hazard zone size (small and large) were used. However, the connection density and population density levels were generated differently than in previous levels of problem instances. First, the connection density levels were generated in three equal ranges from the absolute minimum required for a connected graph, with \( n-1 \) arcs, to a maximum of \( 4n \) arcs, which better mimics real world applications where there is typically at most four-way intersections at corresponding nodes. Therefore, this randomly generated real world abstracted type of problem instances constitute a subset of the low connection density level from the previous type of problem instances, which is broken into a more realistic segmentation of this connection density level \((n-1, 4n)\). Furthermore, the population density levels were assigned in three levels with similar distributions as in the second problem instance type (low: 75% small and 25% large, medium: 50% small and 50%, large: 25% small and 75% large). The small population ranged from 1 to 10,000 and the large from 10,001 to 20,000. The genetic algorithm and Cova heuristic were then used to solve these problem instances and were compared to one another based upon time to find a recommended solution and feasible solutions that had a higher evacuation difficulty.
The fourth problem instance type was a problem instance that was extracted from an actual real world network of eastern Arkansas. The genetic algorithm and Cova heuristic were used to solve this real world problem instance. The time to find a recommended solution and relative evacuation difficulty were then discussed.

For all four problem instance types, the Cova-Church heuristic was used to solve each of the 10 replications of the problem instances with alpha of 0.775 and 128 starts, which was recommended by Dr. Cova (1997) from his experimentation. For this analysis, the time to completion and the final solution with corresponding evacuation difficulty for each trial were recorded. Subsequently, the genetic algorithm was then used to solve each of the 10 replications of problem instances. First, the setting of consecutive non-improving iterations for the early termination criterion was determined for each node level through some pilot tests with the maximum number of iterations allowed set at 7000 for the genetic algorithm in all problem instances. Once a reasonable setting for this was determined, the termination criterion was then set to that level for testing. The time to completion and the final solution with corresponding evacuation difficulty for each trial was recorded for all four problem instance types.

4.3.4. Evaluation Results

First Problem Instance Type

Validation of Genetic Algorithm and Cova Heuristic

For the initial validation of the genetic algorithm and the Cova heuristic, the example problem created to illustrate the Cova heuristic was solved using both. Both the genetic algorithm and Cova heuristic found the optimal solution for this 10 node problem instance 100% of the time. Furthermore, the example problem was then slightly
modified, which is shown in Figure 9, with populations shown at the nodes so that the reachability aspect for a contiguous hazard zone could be tested. From Figure 9, it can be seen that the optimal solution will have to be one of the node clusters of two 100 population nodes on either the right or left. However, all four of the 100 population nodes cannot be included in the hazard zone, which was restricted to five nodes, because this would create a noncontiguous hazard zone. The genetic algorithm ended with the one of the recommended solutions of the two 100 population nodes on the right side of Figure 9. The populations were then varied to further ensure validity of the genetic algorithm and Cova heuristic with all final solutions appropriate for the varied problem instance.

Thus, the code for the Cova heuristic and the genetic algorithm were validated to be coded correctly. For further validation and analysis of the genetic algorithm and the Cova heuristic code, the experimental trials for the 20 node problem instance type were also conducted.

20 Node Problem Instances

First, pilot test trials were run to determine the appropriate termination criterion, which was the number of consecutive non-improving iterations. For the genetic algorithm (GA)
as specified previously, non-improving was defined as having less than a 0.01% gain in the fitness function (evacuation difficulty) of a contiguous feasible hazard zone. For the 20 node level problem instances, both the genetic algorithm and the Cova heuristic were significantly faster at reaching a recommended solution than total enumeration; however, the Cova heuristic was significantly faster than the genetic algorithm. For the pilot test trials, the average time to completion for the Cova heuristic was 3.1421 seconds. As one can see in highlighted selection of Table 3, at 400 consecutive non-improving iterations of the genetic algorithm, there was no improvement in the proportion of trials with greater evacuation difficulties with the genetic algorithm having 0.778 of the trial replications with a greater evacuation difficulty and the proportion of trial replications’ overall time to find a solution was longer than the Cova heuristic in all trial replications from 300 consecutive non-improving iteration termination criterion to 500. Moreover, the genetic algorithm took an average time of 6.8192 seconds longer to find a solution than the Cova heuristic. Thus, the early termination criterion of 400 consecutive non-improving iterations for the 20 node level problem instances was used due to the fact that with the average additional run time of approximately two seconds, the genetic algorithm found the optimal solution 9.3% more often than with 200 and 300 consecutive non-improving iterations.
Once the appropriate early termination criterion was set at 400 consecutive non-improving iterations, the 10 replications of all combinations of the three variables of interest (population density, connection density, and maximum hazard zone size) were run. This consists of 180 trials for both the Cova heuristic and genetic algorithm. From these trials, the GA took an average of 9.9614 seconds with a 95% confidence interval of (9.7372, 10.1855) seconds for the 400 consecutive non-improving iteration termination criteria, the Cova heuristic took an average of 3.1421 seconds with a 95% confidence interval of (2.8342, 3.4501) seconds, and the total enumeration code took an average of 155.6731 with a 95% confidence interval of (145.2641, 166.0491) seconds.

Furthermore, the gray highlighted section of Table 4 shows that the genetic algorithm terminated at the optimal solution more than 50% of the time with respect to all connection density, population density, and maximum hazard zone size levels. The high population density level had the lowest proportion reaching the optimal solution with a 95% confidence interval of (0.603, 0.831), and the proportion of trials that recommend the optimal solution for medium connection density level had the highest proportion of trials with the optimal solution with a 95% confidence interval of (0.760, 0.940) and an
overall proportion of trials reaching the optimal solution with a 95% confidence interval of (0.735, 0.853).

Table 4: Comparison of Genetic Algorithm ED to Optimal ED

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA ED equal to Optimal) Overall</td>
<td>0.735</td>
<td>0.794</td>
<td>0.853</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) PopDensL</td>
<td>0.760</td>
<td>0.850</td>
<td>0.940</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) PopDensM</td>
<td>0.719</td>
<td>0.817</td>
<td>0.915</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) PopDensH</td>
<td>0.603</td>
<td>0.717</td>
<td>0.831</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) ConnDensL</td>
<td>0.679</td>
<td>0.783</td>
<td>0.888</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) ConnDensM</td>
<td>0.760</td>
<td>0.850</td>
<td>0.940</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) ConnDensH</td>
<td>0.640</td>
<td>0.750</td>
<td>0.860</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) HzS</td>
<td>0.770</td>
<td>0.844</td>
<td>0.919</td>
</tr>
<tr>
<td>Proportion(GA ED equal to Optimal) HzL</td>
<td>0.654</td>
<td>0.744</td>
<td>0.835</td>
</tr>
</tbody>
</table>

In Table 5, the Cova heuristic terminated at the optimal solution less than 50% of the time with respect to all connection density, population density, and maximum hazard zone size levels. The large hazard zone level had the lowest proportion reaching the optimal solution with a 95% confidence interval of (0.108, 0.270). The proportion of trials that recommend the optimal solution for small hazard zone level had the highest proportion of trials with the optimal solution with a 95% confidence interval of (0.236, 0.431) and an overall proportion of trials reaching the optimal solution with a 95% confidence interval of (0.197, 0.325).
Table 5: Comparison of Cova Heuristic ED to Optimal ED

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(Cova ED equal to Optimal) Overall</td>
<td>0.197</td>
<td>0.261</td>
<td>0.325</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) PopDensL</td>
<td>0.140</td>
<td>0.250</td>
<td>0.360</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) PopDensM</td>
<td>0.169</td>
<td>0.283</td>
<td>0.397</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) PopDensH</td>
<td>0.140</td>
<td>0.250</td>
<td>0.360</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) ConnDensL</td>
<td>0.199</td>
<td>0.317</td>
<td>0.434</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) ConnDensM</td>
<td>0.184</td>
<td>0.300</td>
<td>0.416</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) ConnDensH</td>
<td>0.072</td>
<td>0.167</td>
<td>0.261</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) HzS</td>
<td>0.236</td>
<td>0.333</td>
<td>0.431</td>
</tr>
<tr>
<td>Proportion(Cova ED equal to Optimal) HzL</td>
<td>0.108</td>
<td>0.189</td>
<td>0.270</td>
</tr>
</tbody>
</table>

In Figure 10, one can see the convergence plot of one test trial for the 20 node level. As shown in the graph, the genetic algorithm starts at a solution that has lower evacuation difficulty and the genetic algorithm pushes for a better solution to the problem instance until the solution converges to the final recommended solution at which the termination criteria of either the maximum number of iterations or number of consecutive non-improving iterations has been reached. For this example experimental trial, the genetic algorithm terminated at 433 iterations. The maximum number of iterations, which was set at 7000 iterations, was never reached and terminated early due to consecutive non-improving iterations.
From the 20 node level problem instance trials, one can conclude that the Cova heuristic finds a solution significantly faster than the genetic algorithm; however, the genetic algorithm found the optimal solution a significantly higher proportion of the time. The Cova heuristic performed best when the hazard zone size was small because this would significantly reduce the feasible solution space. The large hazard zone size was where the Cova heuristic performed the worst because of the much larger feasible solution space. The second class of problem instances was examined next.

*Second Problem Instance Type*

*32 Node Problem Instances*

For the second problem instance type, the purpose of the analysis was to use the Cova heuristic and genetic algorithm to solve randomly generated connected graphs that cover...
the entire range of possible connection density levels from $n-1$ arcs to $(n*(n-1))/2$. First, pilot test trials were run to determine the appropriate termination criterion, which was the number of consecutive non-improving iterations. For the genetic algorithm as specified previously, non-improving was defined as having less than 0.01% gain in the fitness function (evacuation difficulty) of a contiguous feasible hazard zone. For the 32 node level of this problem instance type, the genetic algorithm’s proportion of faster overall time to find a solution was compared to the Cova heuristic’s overall time along with the relative proportion of recommended solutions with greater evacuation difficulties due to the fact that total enumeration solution was not available.

For the test trials, the average time to completion for the Cova heuristic was 17.349 seconds. As one can see in the highlighted selection of Table 6, at 200 consecutive non-improving iterations of the genetic algorithm there was no improvement in the proportion of greater evacuation difficulties. The genetic algorithm had 0.926 of trial replications with a greater evacuation difficulty and the proportion of trial replications’ overall time to find a solution decreased significantly from 0.556 with the additional consecutive non-improving iteration termination criterion. Moreover, the genetic algorithm took an average of 2.195 fewer seconds to find a solution than the Cova heuristic. Therefore, the termination criterion of 200 consecutive non-improving iterations for the 32 node level problem instances was used.
Table 6: Pilot Test Trials for Early Termination Criterion Determination (32 Nodes)

<table>
<thead>
<tr>
<th>Criteria of Interest</th>
<th>Iterations of Equal Best Value Before Termination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Average Time</td>
<td>10.055</td>
</tr>
<tr>
<td>Average Time Difference</td>
<td>-7.294</td>
</tr>
<tr>
<td>GA Overall Time Better</td>
<td>0.685</td>
</tr>
<tr>
<td>GA ED Greater</td>
<td>0.833</td>
</tr>
</tbody>
</table>

Once the appropriate termination criterion was set, the 10 replications of all combinations of the three variables of interest (population density, connection density, and maximum hazard zone size) were run. This amounted to 180 trials for both the Cova heuristic and the genetic algorithm. From these trials, it can be seen in the green (cross-hatched) section of Table 7 that the genetic algorithm and Cova heuristic were statistically equivalent with respect to the overall time to find a solution because the 95% confidence interval on the proportion of time that the genetic algorithm was better than the Cova heuristic (0.410, 0.556) includes 0.50. Also in Table 7, it can be seen in the gray highlighted section that the genetic algorithm had a significantly better proportion of finding a solution with a greater evacuation difficulty, which can be seen in the 95% confidence interval (0.754, 0.868) with an average proportion of 0.128 of the two finding equivalent evacuation difficulties.

Table 7: 32 Node Overall Time and ED Comparison

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time greater) Overall</td>
<td>0.410</td>
<td>0.483</td>
<td>0.556</td>
</tr>
<tr>
<td>Proportion(ED equal) Overall</td>
<td>0.079</td>
<td>0.128</td>
<td>0.177</td>
</tr>
<tr>
<td>Proportion(GA ED greater) Overall</td>
<td>0.754</td>
<td>0.811</td>
<td>0.868</td>
</tr>
</tbody>
</table>
From the test trials, it can be seen in the green highlighted sections of Table 8 that the genetic algorithm and Cova heuristic were statistically equivalent with respect to the proportion of trials with a faster time to find a solution for all three levels of population density due to the fact that all the 95% confidence intervals on the proportion of time that the genetic algorithm was better than the Cova heuristic include 0.50. However, the genetic algorithm performed better than the Cova heuristic by finding a solution with a greater evacuation difficulty in all three levels of population density. The 95% confidence intervals in the gray highlighted sections in Table 8 show a low population density of (0.699, 0.901) with an average proportion of 0.133 having final solutions with equivalent evacuation difficulties. The medium population density is (0.679, 0.888) with an average proportion of 0.167 of the solutions with similar evacuation difficulties, and the high population density of (0.760, 0.940) with an average proportion of 0.083 of the solutions with identical evacuation difficulties.

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) PopDensL</td>
<td>0.357</td>
<td>0.483</td>
<td>0.610</td>
</tr>
<tr>
<td>Proportion(GA time better) PopDensM</td>
<td>0.340</td>
<td>0.467</td>
<td>0.593</td>
</tr>
<tr>
<td>Proportion(GA time better) PopDensH</td>
<td>0.373</td>
<td>0.500</td>
<td>0.627</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensL</td>
<td>0.699</td>
<td>0.800</td>
<td>0.901</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensL</td>
<td>0.047</td>
<td>0.133</td>
<td>0.219</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensM</td>
<td>0.679</td>
<td>0.783</td>
<td>0.888</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensM</td>
<td>0.072</td>
<td>0.167</td>
<td>0.261</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensH</td>
<td>0.760</td>
<td>0.850</td>
<td>0.940</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensH</td>
<td>0.013</td>
<td>0.083</td>
<td>0.153</td>
</tr>
</tbody>
</table>

From the test trials, it can be seen in the yellow highlighted section of Table 9 that the Cova heuristic was statistically better than the genetic algorithm with respect to the proportion of trials with a faster time to find a solution for the low
connection density level with a 95% confidence interval (CI) for the genetic algorithm being the faster of (0.024, 0.176). However, it can be seen in the gray highlighted sections of Table 9 for the medium and high connection density levels that the genetic algorithm was statistically better than the Cova heuristic with 95% CIs for the proportion of trials with the faster times of (0.511, 0.755) and (0.603, 0.831) respectively. The intuitive reason for the Cova heuristic finding a solution faster than the genetic algorithm for the lower connection density level was because with a lower connection density, the Cova heuristic had to look at fewer potential nodes to be added into the solution hazard zone and thus took less time to find an overall solution. However, the genetic algorithm performed better than the Cova heuristic in the proportion of trials that had a solution with a better evacuation difficulty in all three levels of connection density as seen in the 95% CIs in Table 9. The trials with a low connection density had a 95% CI of (0.621, 0.845) with an average proportion of 0.183 of solutions with equivalent evacuation difficulties. The medium connection density had a CI of (0.739, 0.953) with an average proportion of 0.083 solutions with evacuation difficulties, and the high connection density had a CI of (0.739, 0.928) with an average proportion of 0.117 of the solutions with the same evacuation difficulties.
### Table 9: Node Time and ED Comparison for Connection Density Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) ConnDensL</td>
<td>0.024</td>
<td>0.200</td>
<td>0.370</td>
</tr>
<tr>
<td>Proportion(GA time better) ConnDensM</td>
<td>0.511</td>
<td>0.633</td>
<td>0.755</td>
</tr>
<tr>
<td>Proportion(GA time better) ConnDensH</td>
<td>0.603</td>
<td>0.717</td>
<td>0.831</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensL</td>
<td>0.621</td>
<td>0.733</td>
<td>0.845</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensL</td>
<td>0.085</td>
<td>0.183</td>
<td>0.281</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensM</td>
<td>0.781</td>
<td>0.867</td>
<td>0.953</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensM</td>
<td>0.013</td>
<td>0.083</td>
<td>0.153</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDenH</td>
<td>0.739</td>
<td>0.833</td>
<td>0.928</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensH</td>
<td>0.035</td>
<td>0.117</td>
<td>0.198</td>
</tr>
</tbody>
</table>

From the test trials, one can see in the yellow highlighted section of Table 10 that the Cova heuristic was statistically better than the genetic algorithm with respect to the proportion of trials with a faster time to find a solution for the small hazard zone level with a 95% CI with the genetic algorithm being the faster of (0.185, 0.370). However, as shown in the gray highlighted sections of Table 10, for the large hazard zone level, the genetic algorithm was statistically better than the Cova heuristic with a 95% CI of (0.593, 0.785). The intuitive reason for the Cova heuristic finding a solution faster than the genetic algorithm for the small hazard zone level is due to the fact that the Cova heuristic will stop once there are no nodes that have a potential gain greater the specified alpha level or once it reaches the maximum hazard zone size allowed. If the maximum hazard zone size is small, then the Cova heuristic will only have to look at adding a small number of nodes to the solution hazard zone and will thus take less time to find an overall solution. However, the genetic algorithm performed better than the Cova heuristic in finding a solution with a greater evacuation difficulty in both levels of hazard zone level as seen in the 95% CIs in Table 10 for the small hazard zone size of (0.630, 0.815) with an average proportion of 0.222 of the solutions with identical evacuation difficulties, and
for the large hazard zone level of (0.838, 0.962) with an average proportion of 0.033 of solutions with similar evacuation difficulties.

Table 10: 32 Node Time and ED Comparison for Maximum Hazard Zone Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) - HzS</td>
<td>0.185</td>
<td>0.278</td>
<td>0.370</td>
</tr>
<tr>
<td>Proportion(GA time better) - HzL</td>
<td>0.593</td>
<td>0.689</td>
<td>0.785</td>
</tr>
<tr>
<td>Proportion(GA ED greater) - HzS</td>
<td>0.630</td>
<td>0.722</td>
<td>0.815</td>
</tr>
<tr>
<td>Proportion(ED equal) HzS</td>
<td>0.136</td>
<td>0.222</td>
<td>0.308</td>
</tr>
<tr>
<td>Proportion(GA ED greater) HzL</td>
<td>0.838</td>
<td>0.900</td>
<td>0.962</td>
</tr>
<tr>
<td>Proportion(ED equal) HzL</td>
<td>0.000</td>
<td>0.033</td>
<td>0.070</td>
</tr>
</tbody>
</table>

In Figure 11, one can see the convergence plot of one test trial for the 32 node level. As shown in the graph, the genetic algorithm starts at a solution that has lower evacuation difficulty and pushes for a better solution to the problem instance until the solution converges to the final recommended solution, where the termination criteria of either the maximum number of iterations or number of consecutive non-improving iterations is reached. For this trial, the genetic algorithm terminated at 361 total iterations. For the entire 32 node level, the maximum number of iterations, which was set at 7000 iterations, was never reached and terminated due to consecutive non-improving iterations.
64 Node Problem Instances

First, pilot test trials were run to determine the appropriate termination criterion, which was the number of consecutive non-improving iterations. For the genetic algorithm, as specified previously, non-improving was defined as having less than 0.01% gain in the fitness function (evacuation difficulty) of a contiguous feasible hazard zone. For the 64 node level of this problem instance type, the genetic algorithm’s proportion of faster overall time to find a solution compared to the Cova heuristic along with the relative proportion of recommended solutions with greater evacuation difficulties due to the fact that total enumeration was not an option time wise. For the test trials, the average time to completion for the Cova heuristic was 325.141 seconds. As one can see in gray highlighted selection of Table 11, at 500 consecutive non-improving iterations of the genetic algorithm, there was no improvement in proportion of greater evacuation difficulties with the genetic algorithm having 0.815 of trial replications with a greater
evacuation difficulty and the proportion of trial replications’ overall time to find a solution decreasing significantly from 0.648 to 0.4426 with the additional consecutive non-improving iteration termination criterion. Moreover, the genetic algorithm took an average of 176.336 fewer seconds to find a solution than the Cova heuristic. When consecutive non-improving iterations were increased to 700, the genetic algorithm took an average of 409.008 seconds longer than the Cova heuristic to find a recommended solution while only finding a proportion of solutions with greater evacuation difficulty of 0.018. Therefore, the termination criterion of 500 consecutive non-improving iterations for the 64 node level problem instances was used.

Once the appropriate termination criterion was set, the 10 replications of all combinations of the three variables of interest (population density, connection density, and maximum hazard zone size) were run. Just as in the 32 node level problem instances, 180 trials were used for both the Cova heuristic and genetic algorithm for 64 nodes. From these trials, one can see gray highlighted sections of Table 12 that the genetic algorithm had a significantly better proportion of overall time to find a solution than the Cova heuristic with a 95% CI of (0.785, 0.893) and a significantly better proportion of trials that found a solution with a greater evacuation difficulty, which can be seen in the 95% CI (0.766,
0.878) with an average proportion of 0.039 of the two finding equivalent evacuation difficulties.

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) Overall</td>
<td>0.785</td>
<td>0.839</td>
<td>0.893</td>
</tr>
<tr>
<td>Proportion(ED equal) Overall</td>
<td>0.011</td>
<td>0.039</td>
<td>0.067</td>
</tr>
<tr>
<td>Proportion(GA ED greater) Overall</td>
<td>0.766</td>
<td>0.822</td>
<td>0.878</td>
</tr>
</tbody>
</table>

From the test trials, it can be seen in the gray highlighted sections of Table 13 that the genetic algorithm was able to find a solution faster than the Cova heuristic in all three population density levels with the low population density level having a 95% CI for the proportion of trials with a faster time of (0.760, 0.940), and both the medium and high population density levels having a 95% CIs for the proportion trials with a faster time of (0.739, 0.928). Furthermore, the genetic algorithm performed better than the Cova heuristic in finding a solution with a greater evacuation difficulty in all three levels of population density as seen in the 95% CIs for low population density of (0.640, 0.860) with an average proportion of 0.033 of solutions having the same evacuation difficulties, for the medium population density of (0.760, 0.940) with an average proportion of 0.050 of solutions with similar evacuation difficulties, and for the high population density of (0.781, 0.953) with an average proportion of 0.033 of solutions with similar evacuation difficulties.
Table 13: 64 Node Time and ED Comparison for Population Density Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion (GA time better) PopDensL</td>
<td>0.760</td>
<td>0.850</td>
<td>0.940</td>
</tr>
<tr>
<td>Proportion (GA time better) PopDensM</td>
<td>0.739</td>
<td>0.833</td>
<td>0.928</td>
</tr>
<tr>
<td>Proportion (GA time better) PopDensH</td>
<td>0.739</td>
<td>0.833</td>
<td>0.928</td>
</tr>
<tr>
<td>Proportion (GA ED greater) PopDensL</td>
<td>0.640</td>
<td>0.750</td>
<td>0.860</td>
</tr>
<tr>
<td>Proportion (ED equal) PopDensL</td>
<td>0.000</td>
<td>0.033</td>
<td>0.079</td>
</tr>
<tr>
<td>Proportion (GA ED greater) PopDensM</td>
<td>0.760</td>
<td>0.850</td>
<td>0.940</td>
</tr>
<tr>
<td>Proportion (ED equal) PopDensM</td>
<td>0.000</td>
<td>0.050</td>
<td>0.105</td>
</tr>
<tr>
<td>Proportion (GA ED greater) PopDensH</td>
<td>0.781</td>
<td>0.867</td>
<td>0.953</td>
</tr>
<tr>
<td>Proportion (ED equal) PopDensH</td>
<td>0.000</td>
<td>0.033</td>
<td>0.079</td>
</tr>
</tbody>
</table>

From the test trials, it can be seen in the green highlighted sections of Table 14 that the Cova heuristic and the genetic algorithm were statistically equivalent with respect to the proportion of trials with a faster time to find a solution for the low connection density level as well as having statistically equivalent proportion of evacuation difficulties for this connection density level. However, one can see in the gray highlighted sections of Table 14 that for both the medium and high connection density levels, the genetic algorithm was statistically better than the Cova heuristic with 95% CIs of (0.895, 1). The intuitive reason for the Cova heuristic finding a solution faster than the genetic algorithm for the lower connection density level is due to the fact that the with a lower connection density the Cova heuristic will have to look at fewer potential nodes to be added into the solution hazard zone and thus take less time to find an overall solution. However, the increase in the number of nodes in the problem instance appeared to cancel this advantage. Moreover, the genetic algorithm performed better than the Cova heuristic in finding a solution with a greater evacuation difficulty in the medium and high connection density levels as seen in the 95% CIs in Table 14 for the medium connection density of (0.824, 0.976) with an average proportion of 0.017 of the solutions with equivalent
evacuation difficulties, and for the high connection density of (0.895, 1) with an average proportion of 0.017 of the solutions with similar evacuation difficulties.

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) ConnDensL</td>
<td>0.494</td>
<td>0.617</td>
<td>0.740</td>
</tr>
<tr>
<td>Proportion(GA time better) ConnDensM</td>
<td>0.895</td>
<td>0.950</td>
<td>1.000</td>
</tr>
<tr>
<td>Proportion(GA time better) ConnDensH</td>
<td>0.895</td>
<td>0.950</td>
<td>1.000</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensL</td>
<td>0.494</td>
<td>0.617</td>
<td>0.740</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensL</td>
<td>0.013</td>
<td>0.083</td>
<td>0.153</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensM</td>
<td>0.824</td>
<td>0.900</td>
<td>0.976</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensM</td>
<td>0.000</td>
<td>0.017</td>
<td>0.049</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensH</td>
<td>0.895</td>
<td>0.950</td>
<td>1.000</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensH</td>
<td>0.000</td>
<td>0.017</td>
<td>0.049</td>
</tr>
</tbody>
</table>

From the test trials, it can be seen in the gray highlighted section of Table 15 that the genetic algorithm was statistically better than the Cova heuristic with respect to the proportion of trials with a faster time to find a solution for both maximum hazard zone levels with 95% CIs of (0.679, 0.854) and (0.852, 0.970) for the small and large hazard zone levels, respectively. Furthermore, the genetic algorithm performed better than the Cova heuristic in finding a solution with a greater evacuation difficulty in both levels of hazard zone level as seen in the 95% CIs in Table 15 for the small hazard zone size of (0.642, 0.825) with an average proportion of 0.078 of the solutions with identical evacuation difficulties, and for the large hazard zone level of (0.852, 0.970) with zero having equivalent evacuation difficulties.
Table 15: 64 Node Time and ED Comparison for Maximum Hazard Zone Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) - HzS</td>
<td>0.679</td>
<td>0.767</td>
<td>0.854</td>
</tr>
<tr>
<td>Proportion(GA time better) - HzL</td>
<td>0.852</td>
<td>0.911</td>
<td>0.970</td>
</tr>
<tr>
<td>Proportion(GA ED greater) - HzS</td>
<td>0.642</td>
<td>0.733</td>
<td>0.825</td>
</tr>
<tr>
<td>Proportion(ED equal) HzS</td>
<td>0.022</td>
<td>0.078</td>
<td>0.133</td>
</tr>
<tr>
<td>Proportion(GA ED greater) HzL</td>
<td>0.852</td>
<td>0.911</td>
<td>0.970</td>
</tr>
<tr>
<td>Proportion(ED equal) HzL</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

In Figure 12, the convergence plot of one test trial for the 64 node level can be seen. As shown in the graph, the genetic algorithm starts at a solution that has a lower evacuation difficulty and pushes for a better solution to the problem instance until a final recommended solution has been reached due to the termination criteria of either the maximum number of iterations or the maximum number of consecutive non-improving iterations. For this example experimental trial, the genetic algorithm terminated at 673 total iterations. For the entire 64 node level, the maximum number of iterations, which was set at 7000 iterations, was never reached and terminated due to 500 consecutive non-improving iterations.
First, pilot test trials were run to determine the appropriate termination criterion, which was the number of consecutive non-improving iterations. For the genetic algorithm as specified previously, non-improving was defined as having less than 0.01% gain in the fitness function (evacuation difficulty) of a contiguous feasible hazard zone. For the 128 node level of this problem instance type, the genetic algorithm’s had a faster proportion of overall time to find a solution compared to the Cova heuristic. The genetic algorithm also had a relative proportion of recommended solutions with greater evacuation difficulties due to the fact that total enumeration was not an option. For the pilot test trials, the average time to completion for the Cova heuristic was 7691.344 seconds. As one can see in gray highlighted selection of Table 16, at 500 consecutive non-improving
iterations of the genetic algorithm there was no improvement in the proportion of greater evacuation difficulties with the genetic algorithm having 94.4% of trial replications with a greater evacuation difficulty. Moreover, the genetic algorithm took an average time of 6849.055 seconds less to find a solution than the Cova heuristic. When consecutive non-improving iterations were increased to 700, the genetic algorithm took an average of 6773.574 seconds less than the Cova heuristic to find a recommended solution with no improvement in the proportion of time to find a recommended solution. Therefore, the termination criterion of 500 consecutive non-improving iterations for the 128 node level problem instances was used.

<table>
<thead>
<tr>
<th>Criteria of Interest</th>
<th>100</th>
<th>300</th>
<th>500</th>
<th>700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Time</td>
<td>391.585</td>
<td>620.924</td>
<td>842.289</td>
<td>917.770</td>
</tr>
<tr>
<td>Average Time Difference</td>
<td>-7299.759</td>
<td>-7070.420</td>
<td>-6849.055</td>
<td>-6773.574</td>
</tr>
<tr>
<td>GA Overall Time Better</td>
<td>0.889</td>
<td>0.889</td>
<td>0.889</td>
<td>0.852</td>
</tr>
<tr>
<td>GA ED Greater</td>
<td>0.667</td>
<td>0.796</td>
<td>0.944</td>
<td>0.944</td>
</tr>
</tbody>
</table>

Once the appropriate termination criterion was set, the 10 replications of all combinations of the three variables of interest (population density, connection density, and maximum hazard zone size) were run. Just as in the 32 and 64 node level problem instances, this amounted to 180 trials for both the Cova heuristic and genetic algorithm to run with 128 nodes. From these trials, one can see gray highlighted sections of Table 17 that the genetic algorithm had a significantly better proportion of trials with overall time to find a solution than the Cova heuristic with a 95% CI of (0.843, 0.935). The genetic algorithm also had a significantly better proportion of trials with a solution with a greater
evacuation difficulty, which can be seen in the 95% CI (0.798, 0.902) with an average proportion of 0.017 of the two finding equivalent evacuation difficulties.

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) Overall</td>
<td>0.843</td>
<td>0.889</td>
<td>0.935</td>
</tr>
<tr>
<td>Proportion(ED equal) Overall</td>
<td>0.000</td>
<td>0.017</td>
<td>0.035</td>
</tr>
<tr>
<td>Proportion(GA ED greater) Overall</td>
<td>0.798</td>
<td>0.850</td>
<td>0.902</td>
</tr>
</tbody>
</table>

From the test trials, it is shown in the gray highlighted sections of Table 18 that the genetic algorithm was able to find a solution faster than the Cova heuristic in all three population density levels with the high population density level having a 95% CI of (0.802, 0.965), and both the low and medium population density levels having a 95% CIs of (0.802, 0.965). Furthermore, the genetic algorithm performed better than the Cova heuristic in finding a solution with a greater evacuation difficulty in all three levels of population density as seen in the 95% CIs for low population density of (0.760, 0.940) with an average proportion of 0.017 of the solutions with identical evacuation difficulties, for the medium population density of (0.719, 0.915) with an average proportion of 0.017 of the solutions with identical evacuation difficulties, and for the high population density of (0.802, 0.965) with an average proportion of 0.017 of the solutions with equivalent evacuation difficulties.
<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) PopDensL</td>
<td>0.802</td>
<td>0.883</td>
<td>0.965</td>
</tr>
<tr>
<td>Proportion(GA time better) PopDensM</td>
<td>0.802</td>
<td>0.883</td>
<td>0.965</td>
</tr>
<tr>
<td>Proportion(GA time better) PopDensH</td>
<td>0.824</td>
<td>0.900</td>
<td>0.976</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensL</td>
<td>0.760</td>
<td>0.850</td>
<td>0.940</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensL</td>
<td>0.000</td>
<td>0.017</td>
<td>0.049</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensM</td>
<td>0.719</td>
<td>0.817</td>
<td>0.915</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensM</td>
<td>0.000</td>
<td>0.017</td>
<td>0.049</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensH</td>
<td>0.802</td>
<td>0.883</td>
<td>0.965</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensH</td>
<td>0.000</td>
<td>0.017</td>
<td>0.049</td>
</tr>
</tbody>
</table>

From the test trials, it can be seen in the gray highlighted sections of Table 19 that for all three levels of connection density, the genetic algorithm was statistically better than the Cova heuristic with 95% CIs for the proportion of time the GA was better than the Cova heuristic of (0.847, 0.987), (0.760, 0.940), and (0.824, 0.976) for the levels of low, medium and high, respectively. Moreover, the genetic algorithm performed better than the Cova heuristic in the proportion of trials with a solution that had a greater evacuation difficulty for all three connection density levels as seen in the 95% CIs in Table 19 for the low connection density of (0.566, 0.801) with an average proportion of 0.017 of the solutions with equivalent evacuation difficulties for the medium connection density of (0.895, 1) with zero having equivalent evacuation difficulties, and for the high connection density of (0.847, 0.987) with an average proportion of 0.033 of the solutions with identical evacuation difficulties.
Table 19: 128 Node Time and ED Comparison for Connection Density Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time greater) ConnDensL</td>
<td>0.847</td>
<td>0.917</td>
<td>0.987</td>
</tr>
<tr>
<td>Proportion(GA time greater) ConnDensM</td>
<td>0.760</td>
<td>0.850</td>
<td>0.940</td>
</tr>
<tr>
<td>Proportion(GA time greater) ConnDensH</td>
<td>0.824</td>
<td>0.900</td>
<td>0.976</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensL</td>
<td>0.566</td>
<td>0.683</td>
<td>0.801</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensL</td>
<td>0.000</td>
<td>0.017</td>
<td>0.049</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensM</td>
<td>0.895</td>
<td>0.950</td>
<td>1.000</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensM</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensH</td>
<td>0.847</td>
<td>0.917</td>
<td>0.987</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensH</td>
<td>0.000</td>
<td>0.033</td>
<td>0.079</td>
</tr>
</tbody>
</table>

From the test trials, it can be seen in the gray highlighted section of Table 20 that the genetic algorithm was statistically better than the Cova heuristic with respect the proportion of trials with faster times to find a solution for both maximum hazard zone levels with a 95% CI of (0.692, 0.864) for small hazard zone levels; whereas, the genetic algorithm found a faster solution in all of the large hazard zone level problem instances. Furthermore, the genetic algorithm performed better than the Cova heuristic in the proportion of trials in finding a solution with a greater evacuation difficulty in both levels of hazard zone level as seen in the 95% CIs in Table 20: for the small hazard zone size of (0.667, 0.844) with an average proportion of trials of 0.033 of the solutions with equivalent evacuation difficulties, and for the large hazard zone level of (0.897, 0.992) with zero solutions with equivalent evacuation difficulties.
Table 20: 128 Node Time and ED Comparison for Maximum Hazard Zone Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) - HzS</td>
<td>0.692</td>
<td>0.778</td>
<td>0.864</td>
</tr>
<tr>
<td>Proportion(GA time better) - HzL</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Proportion(GA ED greater) - HzS</td>
<td>0.667</td>
<td>0.756</td>
<td>0.844</td>
</tr>
<tr>
<td>Proportion(ED equal) HzS</td>
<td>0.000</td>
<td>0.033</td>
<td>0.070</td>
</tr>
<tr>
<td>Proportion(GA ED greater) HzL</td>
<td>0.897</td>
<td>0.944</td>
<td>0.992</td>
</tr>
<tr>
<td>Proportion(ED equal) HzL</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Figure 13 shows the convergence plot of one test trial for the 128 node level. As shown in the graph, the genetic algorithm starts at a solution that has lower evacuation difficulty and pushes for a better solution to the problem instance until a final recommended solution has been reached due to the termination criteria of either the maximum number of iterations or the maximum number of consecutive non-improving iterations. For this example experimental trial, the genetic algorithm terminated at 784 iterations. For the entire 128 node level, the maximum number of iterations, which was set at 7000 iterations, was never reached and terminated due to 500 consecutive non-improving iterations.
For the third problem instance type, the purpose for analysis was to use the Cova heuristic and genetic algorithm to solve randomly generated connected graphs that are more focused toward actual road networks. Therefore, the range of connections density was from $n-1$ arcs, which is the absolute minimum for a connected graph, to $4n$, because most real world networks have at most a 4-way intersection. Furthermore, the population density levels were assigned in three levels with similar distributions as in the second problem instance type (low: 75% small and 25% large, medium: 50% small and 50%, large: 25% small and 75% large); however, the small population ranged from one to 10,000 and the large from 10,001 to 20,000. No pilot test trials were run due to the fact
that early termination criterion for a 128 node problem instance had already been determined from the previous experiment; therefore, 500 consecutive non-improving iterations was used as the early termination criterion and the maximum iterations was set at 7000.

First, 10 replications of all combinations of the three variables of interest (population density, connection density, and maximum hazard zone size) were run. This amounted to 180 trials for both the Cova heuristic and genetic algorithm to run. From these trials, one can see in green highlighted section of Table 21 that the genetic algorithm and Cova heuristic were statistically equivalent with respect to the proportion of time each had solutions with greater evacuation difficulties due to the fact that the genetic algorithm had a 95% CI for having a greater evacuation difficulty that includes 0.50. However, shown in the yellow (cross hatched) highlighted section of Table 21, the Cova heuristic had a significantly higher proportion of trials with faster solution times.

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) Overall</td>
<td>[0.112, 0.167]</td>
<td>0.151</td>
<td>[0.221, 0.271]</td>
</tr>
<tr>
<td>Proportion(ED equal) Overall</td>
<td>0.098</td>
<td>0.150</td>
<td>0.202</td>
</tr>
<tr>
<td>Proportion(GA ED greater) Overall</td>
<td>0.356</td>
<td>0.428</td>
<td>0.502</td>
</tr>
</tbody>
</table>

From the test trials, it can be seen in the yellow highlighted sections of Table 22 that the Cova heuristic was faster in time to find a solution with respect to the proportion of trials with a faster time to find a solution for all three levels of population density. However, one can see in the green highlighted sections of Table 22 that the genetic algorithm and the Cova heuristic were statistically equivalent in finding evacuation difficulties in all three levels of population density due to the fact that the 95% confidence intervals for
each of the population density levels include 0.50, which would mean that 50% of time
the genetic algorithm is better and 50% of the time the Cova heuristic is better.

Table 22: Real World Oriented 128 Node Time and ED Comparison for Population Density Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) PopDensL</td>
<td>0.072</td>
<td>0.167</td>
<td>0.261</td>
</tr>
<tr>
<td>Proportion(GA time better) PopDensM</td>
<td>0.264</td>
<td>0.383</td>
<td>0.506</td>
</tr>
<tr>
<td>Proportion(GA time better) PopDensH</td>
<td>0.060</td>
<td>0.150</td>
<td>0.240</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensL</td>
<td>0.072</td>
<td>0.167</td>
<td>0.261</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensM</td>
<td>0.264</td>
<td>0.383</td>
<td>0.506</td>
</tr>
<tr>
<td>Proportion(GA ED greater) PopDensH</td>
<td>0.060</td>
<td>0.150</td>
<td>0.240</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensL</td>
<td>0.324</td>
<td>0.450</td>
<td>0.576</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensM</td>
<td>0.324</td>
<td>0.450</td>
<td>0.576</td>
</tr>
<tr>
<td>Proportion(ED equal) PopDensH</td>
<td>0.047</td>
<td>0.133</td>
<td>0.219</td>
</tr>
</tbody>
</table>

Also from the test trials, the yellow (cross-hatched) highlighted sections of Table 23 show that the Cova heuristic was faster in time to find a solution with respect to the proportion of trials with a faster time to find a solution for the low and medium connection density levels, which correspond to rural and suburban networks. However, the first green highlighted section of Table 23 shows that the genetic algorithm and the Cova heuristic were statistically equivalent in time to find a solution for the high connection density level, which corresponds to city networks, due to the 95% confidence interval containing 0.50. The intuitive reason for the Cova heuristic finding a solution faster than the genetic algorithm for the lower connection density level is due to the fact that with a lower connection density, the Cova heuristic will have to look at fewer potential nodes to be added into the solution hazard zone and thus less time to find an overall solution. Moreover, the proportion of trials that the genetic algorithm found greater or equivalent evacuation difficulties was statistically equivalent for all three levels.
of connection density. For the medium and high connection density levels, the 95% confidence intervals included 0.50; however, one must look a little deeper for the low connection density level equivalence. If the proportion of trials that the genetic algorithm was greater or equivalent to the Cova heuristic were lumped together, the 95% confidence interval would show that the genetic algorithm was better 50% of the time and the Cova heuristic was better 50% of the time.

Table 23: Real World Oriented 128 Node Time and ED Comparison for Connection Density Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) ConnDensL</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Proportion(GA time better) ConnDensM</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Proportion(GA time better) ConnDensH</td>
<td>0.373</td>
<td>0.500</td>
<td>0.627</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensL</td>
<td>0.214</td>
<td>0.333</td>
<td>0.453</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensL</td>
<td>0.099</td>
<td>0.200</td>
<td>0.301</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensM</td>
<td>0.407</td>
<td>0.533</td>
<td>0.660</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensM</td>
<td>0.035</td>
<td>0.117</td>
<td>0.198</td>
</tr>
<tr>
<td>Proportion(GA ED greater) ConnDensH</td>
<td>0.292</td>
<td>0.417</td>
<td>0.541</td>
</tr>
<tr>
<td>Proportion(ED equal) ConnDensH</td>
<td>0.047</td>
<td>0.133</td>
<td>0.219</td>
</tr>
</tbody>
</table>

Furthermore, from the test trials, the yellow highlighted sections of Table 24 show that the Cova heuristic was faster in time to find a solution with respect to the proportion of trials with a faster time to find a solution for both hazard zone levels. Moreover, the Cova heuristic had a significantly higher proportion of trials with greater evacuation difficulties for the small hazard zone level with an average of 0.678, which can be found from one minus the mean proportion of trials that the genetic algorithm was greater minus the mean proportion of time the two were equal and a 95% CI of (0.581, 0.774). However, for the large hazard zone level, the genetic algorithm had a significantly higher proportion of trials with a larger evacuation difficulty with 95% confidence interval of (0.692, 0.864) as shown in the gray highlighted section of Table 24. The intuitive reason
for the Cova heuristic finding a solution faster than the genetic algorithm for the small hazard zone level is due to the fact that the Cova heuristic will stop after there are no nodes that have a potential gain greater the specified alpha level or once it reaches the maximum hazard zone size allowed. If the maximum hazard zone size is small, then the Cova heuristic will only have to look at adding a small number of nodes to the solution hazard zone and therefore take less time to find an overall solution. Furthermore, with the 128 node problem instances, there are a total of $2^{128}$ or $3.4 \times 10^{38}$ potential solutions for the genetic algorithm to potentially evaluate. With a small hazard zone size, the solution space is significantly reduced and thus harder for the genetic algorithm to find a good feasible solution to the problem. Therefore, the Cova heuristic has an advantage in its mechanics for finding solutions to these hazard zone level problems. However, when the solution space is expanded to a larger hazard zone size, the genetic algorithm appears to perform better than the Cova heuristic in finding a solution with a greater evacuation difficulty due to the fact that the solution space is increased significantly. Once the real world problem instances consisting of 128 nodes was conducted and evaluated, an actual real world network was then solved by using the genetic algorithm and the Cova heuristic.

Table 24: Real World Oriented 128 Node Time and ED Comparison for Max Hazard Zone Levels

<table>
<thead>
<tr>
<th>Proportion of Interest</th>
<th>Lower Bound (CI 95%)</th>
<th>Mean</th>
<th>Upper Bound (CI 95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion(GA time better) - HzS</td>
<td>0.046</td>
<td>0.111</td>
<td>0.176</td>
</tr>
<tr>
<td>Proportion(GA time better) - HzL</td>
<td>0.136</td>
<td>0.222</td>
<td>0.308</td>
</tr>
<tr>
<td>Proportion(GA ED greater) - HzS</td>
<td>0.022</td>
<td>0.078</td>
<td>0.133</td>
</tr>
<tr>
<td>Proportion(ED equal) HzS</td>
<td>0.205</td>
<td>0.300</td>
<td>0.395</td>
</tr>
<tr>
<td>Proportion(GA ED greater) HzL</td>
<td>0.692</td>
<td>0.778</td>
<td>0.864</td>
</tr>
<tr>
<td>Proportion(ED equal) HzL</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
4.4. Application on a Real Problem

For the fourth problem instance type, a graphical representation was extracted from a real world network of eastern Arkansas, which includes the West Memphis, Newport, Walnut Ridge, and Jonesboro areas. First, as shown in Figure 14, the county lines (red), road network (blue), with Thiessen polygons (green), and nodes with aggregated populations (yellow) from the Thiessen polygons were pulled from a geographic information system. The population of each Thiessen polygon is aggregated at each of the nodes. Thus, the road network or arcs between nodes was then used to develop the inputs to be analyzed using the genetic algorithm and the Cova heuristic. This network has 186 nodes with 280 arcs, which gives a connection density of approximately 1.51 (ratio of arcs to nodes) that corresponds with the low connection density of the real world oriented experiments previously conducted. For the purpose of experimentation, two hazard zone levels, 47 and 138, were chosen for analysis just as in previous experiments. These levels were chosen by first dividing the total available range of (1, 185) possible nodes into the two ranges of (1, 92), which would correspond to a small hazard zone range, and (93, 185), which would correspond to a large hazard zone range. The average of each range (rounded to nearest whole number of nodes) was then set as the corresponding maximum hazard zone size.
From this network and specification of maximum hazard zone sizes allowed, the genetic algorithm and the Cova heuristic were then used to determine the hazard zone. For the small hazard zone level, the genetic algorithm found a solution in 3056.564 seconds with an evacuation difficulty of 18329 that has a hazard zone of two nodes (11 and 66) as denoted by the orange highlighted Thiessen polygons in Figure 15; whereas, the Cova heuristic found the same solution but only required 2384.969 seconds to find the solution. For the large hazard zone size, the genetic algorithm found the same solution as in the small hazard zone level (node 11 and 66) in 4018.942 seconds; whereas, the Cova heuristic found the same solution as in the small hazard zone level in 3017.008 seconds. If one were to look at the denser city areas of West Memphis, Newport, Walnut Ridge, and Jonesboro in Figure 15, the selection of the hazard zone may come into question.
Therefore, the evacuation difficulties of these areas were then calculated by hand for comparison and justification of why both the Cova heuristic and genetic algorithm chose the hazard zone that they did. The nodes that define the West Memphis area have a population of 23205 with 3 exit capacities, which would correspond to an evacuation difficulty of 7735. The nodes that define the Newport area have a population of 7284 with 8 exit capacities, which would correspond to an evacuation difficulty of 910.5. The nodes that define the Walnut Ridge area have a population of 9060 with 6 exit capacities, which would correspond to an evacuation difficulty of 1510. The nodes that define the Jonesboro area have a population of 53109 with 7 exit capacities, which would correspond to an evacuation difficulty of 7587. From these extra evacuation difficulty calculations, it can be seen that the suggested hazard zone from the genetic algorithm and Cova heuristic has a much higher evacuation difficulty (18329) than all four of these larger city areas.
Figure 15: Identified Hazard Zone of Final Solution

The reason that the Cova heuristic performed better for this real world scenario hinges on two main aspects of the network. As seen in the previous experimentation, the first issue revolves around the fact that the Cova heuristic performs better when there is a lower connection density in the problem instance. This problem instance has a connection density of approximately 1.51, which is relatively low. Due to the fact that the Cova heuristic only looks at adjacent nodes, which are determined through connectivity as discussed earlier, there would be fewer nodes to be potentially added during each iteration of the heuristic and thus take less time. Therefore, the feasible solution space is significantly reduced due to the low connection density and thus creates
a hindrance for the genetic algorithm due to the fact that the genetic algorithm looks at all solutions (even infeasible ones) to eventually drive to a final best feasible solution.

The second issue revolves around the fact that the worst case evacuation scenario only has two nodes in the hazard zone, and, as seen in the previous experimentation, the Cova heuristic performs better when the hazard zone size is small because it adds nodes one at a time until either no nodes can be added that have a potential gain greater than the specified alpha level or all iterations have been conducted. Thus, with only two nodes in the hazard zone of the final solution, time is significantly reduced due to not having to fully iterate each start. If this problem would have had a final solution hazard zone that was larger than two nodes and/or the connection density would have been greater, then the genetic algorithm would have faired much better against the Cova heuristic with possibly faster solution time and greater evacuation difficulty.

5. SUMMARY AND CONCLUSIONS

When a disaster strikes, emergency planners need to be prepared to handle many important duties, such as directing evacuations and distributing emergency supplies. Therefore, emergency planners rely on decision support systems (DSS), which help them to carry out these duties. To be most effective, a DSS should be integrated with a geographic information system (GIS), which provides analysis for the problems that arise and helps users of the DSS to visualize the situation. In addition, an effective DSS should include simulation models and optimization techniques, especially in the pre-event planning. For instance, a simulation model could be used to determine the fastest method of evacuation so that when a hurricane approaches an area, police can use this method to appropriately direct traffic. Using a detailed review of the literature, this
report illuminates the many issues involved in integrating a GIS into a DSS, creating simulation models, and applying optimization techniques so that emergency planners are well prepared should a disaster occur. In addition, the research examines the application of optimization to an important emergency planning situation (finding hazard zones) to illustrate the importance of these techniques.

In summary, the Cova heuristic (1997) was constructed as a basic search algorithm that starts with a root node then looks at only adding adjacent nodes to the hazard zone of the solution. The potential gain in the objective function for each adjacent node was then calculated and of the nodes with potential gains greater than a specified alpha level, one was randomly selected to be added to the existing hazard zone. The heuristic terminates once the maximum hazard zone size is reached or no nodes with a potential gain greater than alpha were available with the solution that had the highest evacuation difficulty stored and output to the user.

The genetic algorithm developed for comparison to the Cova heuristic consisted of an initial random generation of chromosomes. The fitness of these chromosomes was then evaluated and the fittest 30%, which were ensured to be diverse through the mechanics of the algorithm, were stored. These chromosomes were then used to develop new chromosomes through crossover and mutation. The remaining chromosomes were developed through immigration. Finally, the genetic algorithm terminated once it either reached the maximum iterations allowed or the specified number of consecutive non-improving iterations had been reached with the solution that had the highest evacuation difficulty stored and output to the user.

The main focus of the experimentation was to develop a genetic algorithm that would potentially yield better solutions to node-arc networks with respect to evacuation difficulty and time to find a solution for the purpose of improving the process of
determining the vulnerability of given areas to disasters. The Cova heuristic and the genetic algorithm were both coded in Microsoft Visual Basic; furthermore, all experiments were conducted on computers with identical specifications. Both the genetic algorithm and the Cova heuristic were then tested on four types of problem instances.

The first, second, and third problem instance type were randomly generated. The first problem instance type consisted of a small node set such that total enumeration could be used to solve so that the genetic algorithm and the Cova heuristic could be tested for closeness to optimality as well as time to find a solution. For these problem instances, three levels of connection density, population density, and maximum hazard zone size were randomly generated as well. The second problem instance type consisted of three node levels with three connection densities that ranged from the absolute minimum and absolute maximum number of arcs possible for each node level with the population density and maximum hazard zone size generated in a similar manner as in the first problem instance type. The third problem instance type was also randomly generated; however, the three connection density levels were constructed to mimic real world networks with a slight modification to the range for the three population density levels with the maximum hazard zone size generated in a similar manner as in the previous problem instance types. Finally, the fourth problem instance type consisted of a real world network of eastern Arkansas with two levels of maximum hazard zone size chosen.

For the first problem instance type (20 node level only), the Cova heuristic was able to find a solution faster than the genetic algorithm for every test trial. However, the genetic algorithm was able to find the optimal solution significantly more times than the Cova heuristic for all levels of population density, connection density and maximum hazard zone size. Moreover, the Cova heuristic’s highest proportion of trials that found the optimal solution occurred when the hazard zone size was small.
For the second problem instance type, three node levels (32, 64, and 128) were analyzed. For the 32 node level, the genetic algorithm and the Cova heuristic were statistically equivalent in time to find a solution with the genetic algorithm finding a solution with a greater evacuation difficulty a significantly higher proportion of the time in all levels of connection density, population density, and maximum hazard zone size. However, the low connection density level and small hazard zone level, the Cova heuristic found a solution significantly faster than the genetic algorithm. For the 64 node level, the genetic algorithm was significantly faster in finding a solution in all levels of population density, maximum hazard zone size, and connection density except for the low connection density level, where they were statistically equivalent. Furthermore, the genetic algorithm found a solution with a greater evacuation difficulty a significantly higher proportion of the time in all levels of connection density, population density, and maximum hazard zone size. For the 128 node level, the genetic algorithm was significantly faster in finding a solution in all levels of population density, maximum hazard zone size, and connection density; as well as, the genetic algorithm found a solution with a greater evacuation difficulty a significantly higher proportion of the time in all levels of connection density, population density, and maximum hazard zone size.

For the third problem instance type, more realistic connection density levels were generated with slightly modified population density levels. For these trials, the Cova heuristic had a significantly higher proportion of trials with faster times in all levels of population density, maximum hazard zone size, and connection density except for the high connection density, which was statistically equivalent to the genetic algorithm. The Cova heuristic and the genetic algorithm were statistically equivalent in proportion of trials with greater evacuation difficulties for all levels of population density, connection density, and for the small hazard zone size. However, the genetic algorithm had a
statistically higher proportion of trials with greater evacuation difficulty for the large hazard zone size.

The reason the Cova heuristic performed much better against the genetic algorithm for this problem instance type has to do with an issue found from previous experimentation. As seen in the previous experimentation, the issue revolves around the fact that the Cova heuristic performs better when there is a lower connection density of the problem instance. Since, the real world related problem instance type is basically a subset of the low connection density level of previous experiments; one would expect that the Cova heuristic would perform better with respect to time. Due to the fact that the Cova heuristic only looks at adjacent nodes, which are determined through connectivity as discussed earlier, there would be fewer nodes to be potentially added during each iteration of the heuristic and, thus, take less time. Furthermore, the feasible solution space is significantly reduced due to the low connection density and thus creates a hindrance for the genetic algorithm due to the fact that the genetic algorithm looks at all solutions (even infeasible ones) to eventually drive to a final best feasible solution so the more the feasible space is reduced the harder it is for the genetic algorithm to find a high evacuation difficulty feasible solution.

For the fourth problem instance type, both the genetic algorithm and the Cova heuristic found the same final solution to the real world network; however, the Cova heuristic was able to find this solution faster. Part of the reason for this is due to the same issue that was discussed for the third problem instance type; in addition, to the fact that the worst case evacuation scenario for this problem only has two nodes in the hazard zone, and, as seen in the previous experimentation, the Cova heuristic performs better when the hazard zone size is small because it adds nodes one at a time until either no nodes can be added that have a potential gain greater than the specified alpha level or all
iterations have been conducted. Thus, with only two nodes in the hazard zone of the final solution and a lower connection density level of the network, time is significantly reduced due to not having to fully iterate each start. If this problem would have had a final solution hazard zone that was larger than two nodes and/or a higher connection density, then the genetic algorithm would have fared much better against the Cova heuristic with possibly faster solution time and a solution with a greater evacuation difficulty.

In conclusion, the genetic algorithm and the Cova heuristic both have strengths and weaknesses as discussed throughout this report but both can still offer much to not only academia but to the practical applications as well. Once a tool has been developed that can accurately and efficiently give solutions of worst case evacuation scenarios to a user, then that user can conduct varying levels of analysis, which leads to future research, using this tool to aid in determining worst case evacuation difficulty before a hazard were to ever occur. With that information, the user can then begin to develop vulnerability maps of all areas within a region that consist of labeling roads with the evacuation difficulty associated with the worst case (most people exiting through that node). After the labeling of all roads in the network through an iterative process of worst case evaluation of networks and sub-networks, educated planning can be conducted such that adequate preparation can be done before a disaster occurs that would alleviate some of the issue with emergency evacuation at the time of a disaster.
6. BIBLIOGRAPHY


APPENDIX A (ANNOTATED BIBLIOGRAPHY)


This paper focuses on the amount of literature that is currently in OR/MS journals and the approximate amounts in each of those areas. The literature review part discusses the two stages of response efforts, which are pre-event and post-event, and suggests that by separating objectives for the two stages can only lead to suboptimal solutions. Furthermore, the paper provides a brief discussion of the four programmatic phases of emergency management, which consists of mitigation, preparedness, response, and recovery.


The focus of this paper is the development of a framework for the digital representation of geographical systems that incorporates the fundamentals of GIS, computer simulation, model base management, and spatial decision support systems (SDSS). The paper proceeds to define and explain the required classes, variable structure, constructor agents, implementor agents, interface objects, and sub-models required for the basic construction of an object-oriented GIS and SDSS.


This paper discusses the development and experimentation of a new methodology and applied framework, which is called a very dynamic GIS (VDGIS), for the real-time integration, manipulation and visualization of urban traffic data. The urban traffic data within the developed VDGIS requires a sequence of manipulations that include some predefined preprocessing functions, selection, and derivation of traffic data, and visualization and animation tasks.
This paper primarily focuses on the development of the integration link between GIS software, which in this case is ARC/INFO, and the configurable emergency management and planning simulator (Configurable Emergency Management and Planning System or CEMPS). GIS provides a database of spatial data that is integrated with a simulation model that deals with the dynamic and uncertain nature of the evacuation process.

The CEMPS is a prototype SDSS designed for planning of emergency evacuations that is designed as a micro-simulator, which allows for a detailed modeling of the evacuating entities. The paper discusses the three levels of integration, which are minimum, intermediate and high, with the CEMPS integrated link being of a high level. The CEMPS-link interface consists of the following: 1) the simulator is activated from within ARC/INFO, 2) the dynamic direct communication and interaction is maintained between the simulator and ARC/INFO, 3) the simulator and ARC/INFO run simultaneously, 4) data is directly read from and input to the INFO database from the simulator.

The paper then proceeds to discuss the hurdles of linking the GIS software with simulation software with the main hurdle being that of the compatibility of data formats between the two. Furthermore, three types of communication links are utilized in the prototype, which consist of dynamic, static and activation links. Finally, the CEMPS was able to prove that it is possible to link existing GIS software with a simulation model to produce a SDSS.


This paper discusses the development and implementation of a multi-objective genetic algorithm integrated with a geographical information system. Any spatial related analysis was calculated directly in the GIS, and the information communicated to the genetic algorithm through loose coupling, which utilizes an exchange file. The main aspect of this paper that pertains to the project of our interest is the actual integration of
the GIS with the model. The solutions are generated by the genetic algorithm were written as ASCII files, which the GIS would then perform spatial analysis using. The GIS would then be prompted to read the files and the proposed management activity was linked through a relational key to the attribute table in the GIS.


The main focus of this paper was to build a database using a GIS frame for the purpose of emergency planning. The authors initially began to develop three databases: a disaster/emergency database, which contain a graphic display of the disaster zone; a facilities database, which consisted of important facilities such as schools, medical facilities, hotels, etc.; a resource database, which was never constructed but would have consisted of construction and engineering resources that would support disaster response operations. The whole premise of the project was to develop the databases and overlay one another to determine the regions affected by the disaster and utilized for resources, facilities, etc. The main aspect of this paper that was helpful was some of the basic concepts and efficiencies of GIS.


The focus of this paper was to introduce a conceptual framework for a future study of a systems approach to rural disaster preparedness. The author plans to utilize the Systems Dynamics approach to develop computer simulation models to be used as “what-if” tools. The author plans to first identify and define the main internal challenges of single organizations and eventually relate the organizations to the overall community. The approach is to aid in determining the factors that may have negative impact and limit the planning process. System Dynamics applications will eventually be evolved into causal diagrams and computer simulation models for multiple scenario testing. The main organization for the focus of the study at this point is the hospital setting with the focus on the analysis of hospital surge capacity.

This paper offers a high-level overview of the potential benefits of GIS in emergency management with a main emphasis on the thinking of a disaster as a cycle with several phases. This paper also provides a brief discussion of the basic elements of databases needed for evaluation of an earthquake hazard. Finally, the paper suggests that higher levels of government have data available necessary for disaster management and sharing will be needed with local governments to enable better disaster management and response.


The focus of this paper is to provide a brief discussion for the use of the spatial decision support systems (SDSS) for vehicle routing. This paper provides some basic information concerning three existing software that have incorporated geographic data and network algorithms such as Tolomeo, Georoute, and Transead GIS software.


This paper discusses the needed elements of a database with the disadvantages of acquiring and then retrieving necessary information for use at that required time. The authors conclude at this point two main facts. First, the data required is not generally available; second, even if the data were available, there is no easy way to retrieve it for effective use. For the model the author’s developed, an assumption that the input data for the model will be available is made. The model consists of: 1) identifying the potential disaster with corresponding risk, which is function of the probability that an event will occur; 2) developing the damage overlay (different levels of impact; 3) identify and evaluate facilities; 4) creation of lists of facilities likely to be affected; 5a) creation of a mitigation list of facilities that require attention first; 5b) creation of a response list of facilities that are at risk but do not have allocated resources; 6) creation of a response plan.

This paper discusses some of the problems with emergency response pertaining to the appropriate data (subsurface information, buildings, transportation networks, etc.), which is not usually readily available, and the data sets that should be collected before starting a computer-based search and rescue operation is conducted. This paper also introduces tools called infrastructure management information systems (IMISs), which are powerful new tools designed for disaster management. Furthermore, the author discusses the need for a system that can be shared across local, state, and federal jurisdictions; in addition, the basic characteristics data that would be required for better accessibility of data is discussed.


This paper's main focus is the classification of existing literature (319 articles) with respect to GIS-based multicriteria decision analysis (MCDA). The paper starts with a description of the increase of GIS focused journal articles. The articles are then classified with respect to the geo-information components of the GIS-MCDA methods. Then the articles are classified based upon the generic elements of the MCDA methods, such as nature of evaluation, the number of individuals involved in the decision-making process, and the nature or uncertainties. The geo-information components were broken into three categories: raster versus vector data models; explicitly spatial criteria versus implicitly spatial criteria; explicitly spatial alternatives versus implicitly spatial alternatives. The MCDA components were broken into three categories: multi-objective decision analysis versus multiattribute decision analysis; individual versus group decision-making; decisions under certainty versus decision under uncertainty, which pertains to the amount of information about the decision that is available to the decision-maker.

This paper focuses mainly upon the development of a SDSS for rural land use planning at the management level, which integrates geo-spatial data with land use systems for impact assessment models. The authors use an integrating method that identifies which certain functions of GIS that do not require integration for the development of a spatial land allocation decision support system (LADSS) that consists of three tiers of analysis. First, the GIS is used for a spatial data capture that is used in the SDSS as data points. The subsequent tier consists of land use models and impact assessment modules. The final tier consists of the graphical user interface (GUI) and GA-based land use planning tools, which utilizes genetic algorithms.


This paper looks at two particular instances in which GIS was used with simulation models to address emergency preparedness. The first situation was a model based upon the Federal Highway Administration’s TRAFLO model, which depicts the progress of area evacuations by the changing of conditions associated with the population movement over a traffic network. The model used the GIS for incorporation of existing traffic roadway networks, maps, census data, etc. Furthermore, the GIS was used to display time-variable traffic flow, traffic backups, vehicle speed, etc. The authors used their model, which was based upon an already established model, and incorporated output presented in a graphic map display.


This paper focuses on defining and implementing a group of methodologies applicable to emergency response for a disaster. The methodologies discussed include: address matching (geocoding), which consists of providing spatial coordinates to objects; routing,
which consists of tools facilitating path finding within a network; allocation, which is the process of identifying the maximum coverage that can be achieved given specific constraints; buffering, which is reclassification technique where a polygon around the area represents a certain tolerance; Thiessen polygon, which is used to determine the influence of point data to its proximate region; dynamic segmentation, which consists of events being related to the route system where the segments are not predefined; GPS-GIS, which consists of using global positioning system (GPS) instruments for real time monitoring of emergency response vehicles.


This paper discusses a prototype SDSS for emergency planners, which is called CEMPS (Configurable Emergency Management and Planning System) to aid in the development of their contingency plans for evacuation from disaster areas, should an evacuation be necessary. Three basic types of simulators are discussed. The first discussed is a micro-simulator, which tracks the detailed movement of individual entities in the network being simulated. The second is a macro-simulator, which makes no attempt to track individual entities and focuses on the fluid flow in networks, and the third is meso-simulators, which involve a discrete simulation that tracks the movements of groups of vehicles. The authors proceed to develop a discrete micro-simulator, which is written in C++, that is linked to a GIS (ARC/INFO) to create a dynamic SDSS.


This paper is focused mainly on the use of the GIS for the planning of roads/routes through the Himalayan terrain. The main aspect that this paper provided for our purpose was some basic background information of GIS technology with some description of software packages that aid in route planning such as PATHDISTANCE in ArcGIS and PATHWAY in IDRISI. Moreover, there was some basic information pertaining to vector GIS and raster GIS, which has the least flexibility in the neighborhood pattern as compared to the vector GIS that was useful in our research.

This paper studied the allocation of people to “safe areas”, or open areas that were of certain distances away from structures defined by standards set by the Greek Organization for the Protection from Earthquakes. Basically, the capacity of each area was determined by the size of the area and the estimation that each person would take up 2 m$^2$ of the space within that area. The dynamic allocation model minimized the area traveled from one’s house with safety being considered the primary factor for attractiveness to this area, which was rated as high, medium, or low. The distance was calculated from a CAD drawing of the layout of the city with the spatial distribution of the population estimated the area of the each block. Finally, the model was setup such that a bias in favor of safer areas was incorporated by setting a maximum distance within which they are preferred before people head to less safe areas.


This paper discusses the integration of GIS, simulation models and computer visualization. GIS provides the ability to examine spatial relationships among entities, simulation modeling provides the ability to represent dynamic relationships, and visualization gives the power to reveal patterns and relationships. There are two basic types of coupling categories of integration of GIS with simulation models. First, loose coupling involves the communication between the GIS and simulation model through the use of exchange files. Second, deep coupling links the GIS with the simulation model through the use of a common user interface. This paper also provides a sample of ArcView GIS script. Moreover, this paper discusses a broad overview of the software the authors developed that links a Visual Basic GUI with a simulation that utilized roadway link lengths from a GIS digital database with another software that analyzed the carbon monoxide levels of a given area.

This paper provides a brief overview of other research projects that have utilized a GIS for the purpose of modeling. For the author’s project analysis, the GIS was used to develop spatial datasets such as important locations that include hospitals, fire and police stations, and population distributions. The main focus of this paper was the analysis of a staged exercise in Mackay, Australia. The GIS was unable to provide answers in real-time due to technical constraints such as computer processing power and the size of the databases; however, the main impediments to implementation of GIS for real-time decision making were custodianship and technology transfer.