AGENT-BASED DISCRETE EVENT SIMULATION MODELING AND EVOLUTIONARY REAL-TIME DECISION MAKING FOR LARGE-SCALE SYSTEMS

by

Shengnan Wu

B.S. in Manufacturing Engineering, Tsinghua University, 2002

M.S. in Mechanical Engineering, George Washington University, 2004

M.S. in Industrial Engineering, University of Pittsburgh, 2005

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2008

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Shengnan Wu

It was defended on

December 8, 2008

and approved by

Dissertation Director: Larry J. Shuman, Professor, Department of Industrial Engineering

Dissertation Co-Director: Bopaya Bidanda, Professor, Department of Industrial Engineering

Carey Balaban, Professor, School of Medicine

Bryan A. Norman, Associate Professor, Department of Industrial Engineering

Oleg Prokopyev, Assistant Professor, Department of Industrial Engineering

Ken Sochats, Assistant Professor, School of Information Sciences

AGENT-BASED DISCRETE EVENT SIMULATION MODELING AND EVOLUTIONARY REAL-TIME DECISION MAKING FOR LARGE-SCALE SYSTEMS

Shengnan Wu, Ph.D.

University of Pittsburgh, 2008

Computer simulations are routines programmed to imitate detailed system operations. They are utilized to evaluate system performance and/or predict future behaviors under certain settings. In complex cases where system operations cannot be formulated explicitly by analytical models, simulations become the dominant mode of analysis as they can model systems without relying on unrealistic or limiting assumptions and represent actual systems more faithfully. Two main streams exist in current simulation research and practice: discrete event simulation and agent-based simulation. This dissertation facilitates the marriage of the two. By integrating the agent-based modeling concepts into the discrete event simulation framework, we can take advantage of and eliminate the disadvantages of both methods.

Although simulation can represent complex systems realistically, it is a descriptive tool without the capability of making decisions. However, it can be complemented by incorporating optimization routines. The most challenging problem is that large-scale simulation models normally take a considerable amount of computer time to execute so that the number of solution evaluations needed by most optimization algorithms is not feasible within a reasonable time frame. This research develops a highly efficient evolutionary simulation-based decision making procedure which can be applied in real-time management situations. It basically divides the entire process time horizon into a series of small time intervals and operates simulation optimization algorithms for those small intervals separately and iteratively. This method improves computational tractability by decomposing long simulation runs; it also enhances system dynamics by incorporating changing information/data as the event unfolds. With respect to simulation optimization, this procedure solves efficient analytical models which can approximate the simulation and guide the search procedure to approach near optimality quickly.

The methods of agent-based discrete event simulation modeling and evolutionary simulation-based decision making developed in this dissertation are implemented to solve a set of disaster response planning problems. This research also investigates a unique approach to validating low-probability, high-impact simulation systems based on a concrete example problem. The experimental results demonstrate the feasibility and effectiveness of our model compared to other existing systems.

Keywords: Agent-based Simulation, Discrete Event Simulation, Simulation Validation, Geographic Information Systems, Evolutionary Systems, Real-time Decision Making, Simulation Optimization, Heuristics, Disaster Response, Emergency Medical Services, Situation Awareness.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

Although I am the only author of this dissertation, I feel much indebted to many people for its successful completion. I am deeply grateful for the generous support from my advisors and sincere friends, Prof. Larry Shuman and Prof. Bopaya Bidanda, who have been unending sources of advice and encouragement for my research and life. I also appreciate the insightful guidance of the other committee members, Profs. Carey Balaban, Bryan Norman, Oleg Prokopyev, and Ken Sochats. Without their time and effort, this dissertation could not have been possible.

The support from the University of Pittsburgh Research Fund, especially vice provost George E. Klinzing is also gratefully acknowledged.

I extend thanks to many other people who facilitated me in my fruitful graduate life. Thanks to all of the industrial engineering faculty members and staff for maintaining an enjoyable and productive education and research environment. Thanks to my colleagues and friends in the IE department, in the Visual Information Systems Center (VISC), at the University, and around the world for sharing their unique knowledge and talents with me.

Finally, I wish to express the deepest gratitude to my greatest family. I thank my parents for life and eternal love to me. I thank my parents-in-law for kindly support to us. Specially, I thank my son, Andy, who brings endless happiness and hope to the family. Most of all, I thank my beloved wife and best friend, "*Fish*," who shares my hard times and pleasant moments.

This work is dedicated to them!

1.0 INTRODUCTION

In the real world, large-scale, complex systems such as disaster response systems, financial systems and production systems are dynamic and are subject to frequent changes due to many internal and/or external reasons. Such systems are being studied actively and extensively in both academia and industry because major failures lead to highly undesirable outcomes.

In this research, advanced simulation and optimization techniques are synthesized and applied to study dynamic, evolutionary systems and to improve system behaviors sequentially in real-time. The integrated framework is developed with the application to disaster response planning and management.

1.1 BACKGROUND

Computer simulation is an attractive approach to evaluate real-world systems by means of imitating system operations numerically and computing various performance measures. A validated simulation system is a potentially valuable tool for comparing system alternatives and it can be extended to facilitate decision making processes. Due to the considerable complexity in large-scale operational systems, their resulting simulation models are computationally intensive. The difficulty in computation presents an obstacle to searching for optimal solutions efficiently using simulation evaluations. The overall objectives of this research are (1) to develop a simulation modeling methodology that combines both flexibility in terms of operations and efficiency in terms of computation time and (2) to develop a simulation optimization procedure that reduces computational difficulty and allows timely decision making for large-scale problems in real-time settings.

Figure 1-1 presents several appealing research thrusts and their interactions in the areas of simulation and optimization. There are mainly three domains: simulation modeling, optimization and queueing, and heuristic search and metamodeling. Simulation refers to the application of computer routines to imitate actual system operations and evaluate its performance based upon system responses. There are two major areas under this umbrella: discrete event simulation (DES) and agent-based simulation (ABS). Their intersection makes a powerful method called hybrid simulation that interweaves both operational flexibility and computational efficiency. Optimization and stochastic queueing models have strong analytical flavors. They primarily use various mathematical formulae to describe systems. Simulation models sometimes incorporate such analytical models as internal components to enhance model performance. Heuristics means "to find" in the Greek. It is a huge cluster of methods for seeking good solutions with reasonable computational effort. The solutions identified by heuristics are not necessarily optimal, but hopefully are near optimal. Metamodeling is a statistical approach that analyzes existing system's input and output data, identifies hidden relationships, and utilizes such information to seek better solutions. Due to their computational efficiency, heuristic search and metamodeling techniques have been applied widely to find high-quality solutions to problems in a timely manner. There exists a promising research area that interweaves the above approaches to develop flexible, efficient simulation optimization methodologies, as illustrated by the shaded area in Figure 1-1. The integrated method has good prospects for making high-quality decisions for large-scale, complex operational systems. The detailed literature relevant to the above methods is provided in Chapter 2.0.

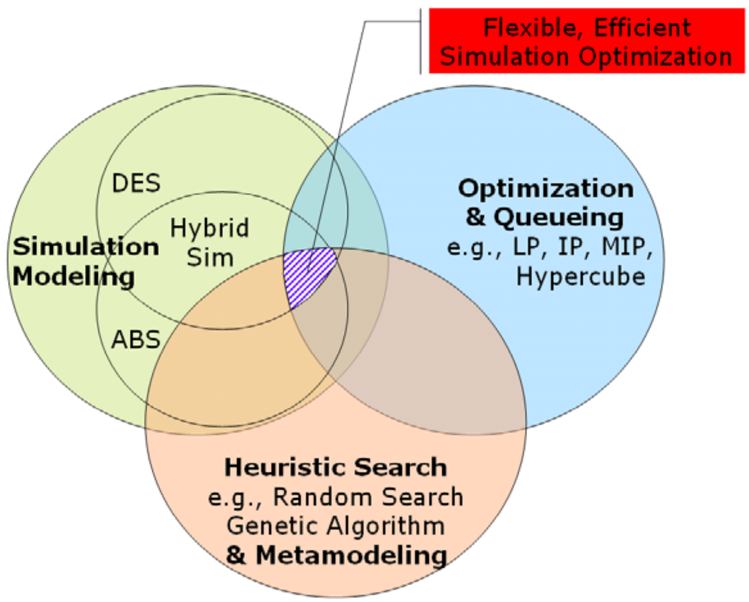

Figure 1-1. Flexible, efficient simulation optimization area

Disaster planning and management serves as a research case for this dissertation. The application is motivated by the following facts:

- Both man-made and natural disaster events are stochastic and hard to control because almost every single event is unique and represents its own specific situation.
- It is not feasible to evaluate the efficiency and effectiveness of different response policies using actual, real disaster events which have great impacts on the society.
- Efficient, comprehensive decision support systems are needed to address real-time disaster decision problems.

Disasters are one of the major barriers to sustainable development of society. Recently, we have observed large-scale natural or man-made disasters that have had great impact on major cities. The catastrophe caused by Hurricane Katrina in New Orleans in 2005 destroyed all aspects of that city including its assets, population and economy. Of the city's 180,000 structures, 125,000 were flooded [30]; one year later, the New Orleans population had been reduced by nearly 60% according to the New York Times [87]. The threat to lives is huge in densely populated urban areas where many structures, facilities and people are concentrated. For a large-scale disaster, even a small delay in responding can exacerbate costs in terms of human lives and property. For many historical disasters, management was impaired because of the lack of pre-event planning and/or proper prediction of the events. The mismanagement of Katrina responses cost more than $100 billion and over 1,300 lives [19]. Improper handling of a disaster not only means delays in responding but also includes overreaction and more-than-needed responses. Thus, how to make the right decisions and respond properly to disasters is a significant question which needs to be researched.

1.2 RESEARCH PROBLEMS AND METHODOLOGIES

In this section, the generic problems and corresponding methodologies developed by the dissertation research are stated. These approaches can be utilized to solve a broad range of problems, although they are applied here to a specific case problem – disaster response planning and management. This section provides an overview of the problem. Each of these components will be discussed in depth in later chapters.

1.2.1 Hybrid simulation

Simulation models have been used widely in studying sophisticated, dynamic system behaviors in both academia and industry. Compared to analytical models, simulations have some favorable advantages in terms of modeling in great detail and capturing the stochastic nature of complex systems without establishing unrealistic assumptions. Discrete event simulation (DES) is a traditional tool for modeling operational systems. In DES, the operation of a system is discretized into a chronological sequence of events. The system state is updated instantly when an event occurs [96]. Recently, agent-based simulation (ABS) has become prevalent in simulation practices. In ABS, entities are modeled as autonomous decision-making individuals who can assess the situation and make their own decisions according to pre-defined rules [9]. ABS is able to simulate the simultaneous operations of multiple agents so it can capture more dynamic interactions in the system.

With the increasing need for system and data integration, simulation models are now required to be interfaced with many other components such as Geographic Information Systems (GIS) to form hybrid platforms instead of standing alone. When various components are interoperated in a seamlessly integrated platform, they can produce high-quality, realistic representations of real-world operational systems.

With regard to implementations, ABS needs to check the system and agents' status much more frequently than DES does. In this sense, ABS is more sensitive to outside environmental changes so it can be integrated with other interactive components more closely. Although ABS is prominent in modeling operational details, it has to sacrifice a large amount of computation time as a tradeoff. Computation time is the bottleneck and concern for most simulation-based studies, especially in the area of simulation-based optimization where many scenarios need to be evaluated by simulation in order to obtain a near-optimal solution. There is a great need to reduce the computation time while maintaining the simulation quality as much as possible. The marriage of DES and ABS (i.e., hybrid simulation) provides the opportunity to achieve this goal. Compared with the existing approaches to combining DES and ABS [35, 68, 108], this dissertation develops a unique modeling data structure for network-centric simulation to reduce the model size and improve efficiency to the greatest extent. The integration method makes the whole simulation system more scalable and facilitates further integration of other components. Also, several specific ideas for incorporating continuous-time models into a discrete-event framework are discussed for a concrete problem. This compact modeling approach can be applied to many problems and areas where network-centric models and interactive agents are involved such as supply chain management, military operations and social network studies.

1.2.2 Rare-event simulation validation

In contrast to analytical models, simulations can represent complex systems better as they imitate actual system operations and measure performance from outcomes directly. However, complex systems normally involve numerous internal logic and rules which are hard to track. For the purpose of validation, simulations are treated as "black-box" systems which can be studied by specifying the inputs and observing the outputs. The good news for most simulation studies is that a wide variety of system response data can be obtained through simulation experiments because many assumptions of analytical models can be relaxed and simulation models are applicable to a broader range of scenarios. Simulation-based experimental data can be compared against actual system statistics to validate the simulation system. Validation is a necessary step for building a reliable simulation model and it has attracted a lot of research interests in the past years. The existing simulation validation methods are reviewed in subsection 2.3.4.

Unfortunately, well-structured validation techniques for rare-event simulations have not been reported widely in the literature. For the purpose of maintenance or preparedness, some low-probability, high-impact events need to be simulated and analyzed beforehand. Those events might have happened rarely, if ever, but may potentially occur in the future and could have great impacts (e.g., dangers leading to major failures) to the operational system under study. Due to the huge potential impacts, physical experiments on the system cannot be conducted. Thus, the actual system data and statistics to compare against the simulation results are not available. Even if some data could have been collected from past events, most of the data would be retrospective and passive. In other words, if data were collected after an event occurred, that data might not truly represent the situation as it developed during the actual event. Thus, using such historical data may flaw the studies.

To circumvent the missing-data situation, the rare-event simulation needs to be validated from different angles including component and system perspectives. This dissertation develops a comprehensive scheme for rare-event simulation validation which is unique in the current literature.

1.2.3 Situation awareness

Situation awareness has been defined formally as the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [25, 26, 27]. Simulation has the ability to predict "future" situations with the "current" states as input. The quoted "future" and "current" indicate the sequence in the time series: “future” is sometime after “current”. According to the definition, the simulation model itself can be regarded as a tool for achieving situation awareness. The application of the concept of situation awareness in system simulation is an innovation in this dissertation.

When modeling a complex system, one can hardly avoid considering the system’s evolution. Human society has evolved over one hundred centuries from hunters and gathers to the modernized world. The world is always changing with time. Real-world events have the common trait of evolutionary development. Normally, the development of complex systems is based on tremendously complicated factors and interactions in a long time horizon. Human beings cannot always perceive the situation well even with sophisticated models because of many unknown factors and limitations. For example, the information acquired may be incomplete and misleading and unexpected situations may suddenly arise to change the outcomes. In the same sense, human decisions should be dynamic in accordance with changing situations.

The evolutionary nature of dynamic systems does not receive enough attention by modelers for many simulation models. This dissertation develops a new evolutionary simulation procedure and incorporates efficient optimization algorithms to obtain a stream of near-optimal solutions over time in order to improve the overall system performance. It basically divides the whole time series into a set of sequential and consecutive small time intervals, simulates and makes decisions iteratively based on the predictions and evaluations in a short time horizon. The system can reduce computational difficulty by running small-sized models and improve simulation quality by incorporating dynamic input data which adapt to the evolving situations.

The evolutionary simulation approach has a broad range of applications in real-time management of complex systems. Some examples of the applicable problems are military operations, disaster responses, emergency room (ER) diagnoses and urban planning. Such problems should have the following characteristics:

- Sequential information input.
- Incomplete information in the beginning, but improves with time.
- Dynamic in nature; new situations may emerge in the middle of the event.
- Optimal streams of decisions.

1.2.4 Real-time simulation optimization

Simulation is a powerful descriptive tool to evaluate system performance. Combined with optimization methods, simulation can be utilized to provide high-quality, robust decisions for system operations. Simulation optimization is analogous to the process of searching for a key in one's pocket: insert one hand into the pocket, feel the likely keys inside and pick the right one out. The likely keys are comparable to the plausible situations which we can be found using simulation and the right key is comparable to the best solution we can obtain through optimization. Simulation-based optimization (simulation optimization for short) has become an active research area. While a simulation model is developed to incorporate dynamic information and evolving situations as described in the last subsection, the decisions should also be dynamic in accordance with changing situations.

Many simulation-based optimization techniques have been developed, e.g., scatter search [38] and surrogate search [62], but few place emphasis on the evolutionary perspective of decision making for dynamic systems. Furthermore, a substantial number of simulation evaluations (e.g., dozens to hundreds of runs per iteration) are normally required to obtain a satisfactory solution. This has not been feasible in real-time decision making cases where large-scale, expensive simulations are involved.

This dissertation develops efficient simulation optimization algorithms to assist in making high-quality, timely decisions and managing online systems. The algorithms utilize analytical formulations and offline experimental results to guide the online search and provide near-optimal solutions quickly. The methodology contributes to the area of simulation optimization and has many potential applications for the limited-resource decision problems.

1.3 SPECIFIC CASE STATEMENT

In this dissertation, the methodologies described in the previous section are applied to a specific problem – simulation-based disaster response planning and management. The overall objective of the application is to provide a circumstance-independent laboratory for testing how the type and scale of the event, situational state, and command decisions affect responders' efficiency and effectiveness in dealing with complex, evolving disasters.

Disasters can be categorized into several major types: natural events, technological events, and human events [24]. Different disasters have distinct characteristics in terms of scale, complexity and treatment, so they require responders to act differently according to specific situations. How to respond to a disaster appropriately is a major challenge for emergency decision makers, e.g., incident managers.

Normally, the emergency managers are professional personnel who have an extensive working knowledge of disaster responses. But to some extent, all disasters are ad-hoc events and they always need special treatment because unthinkable situations can emerge. For example, the

September-11 terrorism attack in New York City is different from Hurricane Katrina in New Orleans so the responses are different. Good disaster decisions are based upon a large amount of information and knowledge of the event. However, because of the complexity of real-life events, human knowledge and experience may not be sufficient to predict how future situations might arise and evolve. Therefore, a computer-based, seamlessly-integrated information sharing and decision support system becomes necessary. It could be used as a tool to help process comprehensive information and make decisions on allocating current resources and dispatching responders to treat the disaster in an appropriate way.

Elegant analytical models can provide quick solutions to complex systems but normally have to build in numerous, sometimes unrealistic assumptions in order to simplify the problem and put it into a feasible mathematical form. In contrast, simulation models can eliminate many of the assumptions by replicating actual processes. For example, a large number of stochastic models basically assume Poisson processes in many places while the simulated operations do not have to follow any particular probability distribution, similar to what occurs in the real world. Therefore, simulation is advantageous in modeling complex, large-scale systems accurately.

Nothing in the world is static. Evolution is literally defined as "a process of change in a certain direction [79]," which is a common characteristic for most complex systems. Any system is rooted in its surrounding, changing environment and interacts with other entities and factors, all of which affect the system iteratively. Thus, a system's status, performance and operations should be altered as time elapses. The problem we are interested in solving is how we can actively change decisions in order to obtain the best overall system performance.

In this research, we develop a disaster response simulation system to achieve the goal of making better decisions based on more realistic models for various disaster scenarios. The system is named the *Dynamic Discrete Disaster Decision Simulation System* (D^4S^2). Unlike many other disaster decision systems (e.g., [12, 47, 114, 128]), D^4S^2 is a comprehensive hybrid simulation system that synthesizes several interactive components including a geographic information system (GIS) and a response rule base to make the whole system more dynamic and realistic. Normal emergency call responses are also considered at the same time as the major disaster occurs. We attempt to validate the computer simulation of such low-probability, high-impact events as disasters, in several ways, some of which are unique in the literature. Using a validated simulation system, the decision makers are able to predict the effects of various critical decisions before actually implementing them at the actual scene. The system can help detect inappropriate decisions early to avoid worsening the situation. In this approach, the responses are revised whenever necessary based on the simulation feedback as the event evolves. Traditionally, simulation is a tool for analyzing and evaluating a complex system's operations. In this decision system, simulation will be used in an innovative manner: it is essentially a dynamic decision driver. The D^4S^2 system provides decision makers with an active laboratory to test policies, strategies and tactics in a simulated real-life decision environment. Thus, it potentially has a wide variety of other applications including but not limited to emergency response planning, military base management and homeland security issues.

1.4 CONTRUBUTIONS

This dissertation significantly contributes to several research areas including both general simulation and optimization methodologies and disaster response applications. First, a modeling methodology is developed to hybrid agent-based and discrete event simulations as well as other information and decision modules into one integrated platform. The research focuses on enhancing the efficiency and scalability of hybrid complex systems by designing a unique model structure for network-centric models. The data structure facilitates the construction of flexible-rule simulations and the efficient combination of discrete-event and time-continuous models.

Second, rare-event simulation validation methods are explored comprehensively and a unique theory based validation is proposed and implemented to validate D^4S^2 from different angles when actual system experiments are impossible.

Third, an evolutionary simulation procedure is developed to strengthen dynamic situation awareness. Many simulation systems do not emphasize the dynamic characteristic of complex systems although some of them allow the simulator to interface/interact with external modules to some extent. When a complex system evolves over a long period of time, the sudden situations that arise cannot be predicted or prepared beforehand. Our procedure is capable of handling the unexpected situations when the simulation is used in real time so as to enhance situation awareness.

Fourth, efficient simulation optimization algorithms are developed to incorporate analytical models, offline experimental results and random factors to obtain near-optimal solutions quickly for the management of complex systems in real time. With this method, the number of needed expensive simulation evaluations is significantly reduced while the solution quality is maintained at satisfactory levels.

Last but not least, the dissertation utilizes the simulation tool to provide some insights into several disaster response and emergency medicine issues, e.g., victim degradation. The approach and results supplement the current medical literature.

1.5 OVERVIEW OF THE DISSERTATION

In this dissertation, Chapter 2.0 provides a summary of the literature about generic decision making methodologies as well as specific applications developed for disaster planning and other complex problems. The first section in this chapter focuses on qualitative methods for disaster management and planning. The following sections introduce quantitative methods including analytical modeling, simulation modeling and simulation-based optimization.

Chapter 3.0 gives an introductory description of the Dynamic Discrete Disaster Decision Simulation System (D^4S^2) that is developed in this work. Typical operations in emergency response systems and the architecture of the integrated computer-based decision support system – D^4S^2 – are presented. The drawbacks of a hard-coded discrete event simulation model are stated which motivate the creation of a more flexible and integrated model.

Chapter 4.0 concentrates on the simulation modeling methodology to combine agent-based models and discrete event simulation. This integration framework is applied to simulate disaster response systems. Specific issues regarding disaster responses such as victim degradation and disaster scene congestion are addressed. As a critical step of a complete simulation study, computational results for validation purposes are then shown to "prove" the correctness of the model.

Chapter 5.0 develops a simulation-based metaheuristic optimization algorithm called an Evolutionary Real-time Decision Making Procedure. The general framework is described, followed by some details regarding the implementation of the procedure such as time parameter selection and analytical modeling methods. An enhanced sub-procedure – *Analytically Guided Randomized Search* (AGRS) – is then presented. Finally, the broad class of problems where the procedure can be applied is summarized.

Chapter 6.0 demonstrates how the simulation-based metaheuristic procedure we develop is applied to the disaster response decision problem by showing the construction of the approximate analytical model and the process of linearization. Comprehensive computational results are provided to confirm the effectiveness and efficiency of the procedure.

The last chapter presents the summary and conclusions for the dissertation by laying out major research contributions and future research directions.

2.0 LITERATURE REVIEW

With the advancements in computer technology and complexity of problems, simulation-based stochastic system modeling and optimization have attracted an increasing amount of interests from both academia and industry. This chapter reviews some key literature related to simulation modeling and optimization as well as the concrete problems of emergency responses where the methods have been applied. Section 2.1 discusses the general methods for disaster management and planning with the focus on qualitative methods and guidelines. Section 2.2 describes the analytical methods used to solve the emergency management problems including mathematical programming and queueing theory. Although the analytical methods can provide quick solutions and good insights, they have more limitations on modeling complex systems in comparison with the simulation-based approaches. Section 2.3 reviews some discrete-event and agent-based simulation models and their combinations. Simulation validation methods are also included because this is crucial in all simulation studies. Section 2.4 discusses the important simulation-based optimization methodologies and applications. Finally, section 2.5 presents a summary of conclusions drawn from reviewing the literature.

2.1 EMERGENCY MANAGEMENT AND PLANNING

Emergency management and planning (or disaster management and planning) is the discipline of dealing with and avoiding risks [51]. It has a comprehensive spectrum including mitigation of potential risks, response to ongoing disasters, recovery after disasters, preparedness to future emergency situations and communications before, during and after disasters.

Emergency management and planning involves a broad class of knowledge and practices. The Homeland Security Council—in partnership with the Department of Homeland Security (DHS), other Federal departments and agencies, and State, local, tribal, and territorial governments—developed the National Planning Scenarios [20]. The Scenarios include various types of emergencies/disasters for both natural and man-made catastrophes across the all-hazards spectrum. They are used as a reference by all levels of governments, agencies and research institutions to explore the consequences and responses of major disasters. The 15 Planning Scenarios are listed in Table 2-1.

Table 2-1. National Planning Scenarios [20]

Improvised Nuclear Device	Major Earthquake
Aerosol Anthrax	Major Hurricane
Pandemic Influenza	Radiological Dispersal Device
Plague	Improvised Explosive Device
Blister Agent	Food Contamination
Toxic Industrial Chemicals	Foreign Animal Disease
Nerve Agent	Cyber Attack
Chlorine Tank Explosion	

From the practice side, a significantly large amount of operations are still driven and regulated by qualitative protocols, standards and policies at different levels, national and local. Some examples of the national agencies, organizations and framework are National Fire Protection Association (NFPA), National Incident Management System (NIMS), Federal Emergency Management Agency (FEMA), and National Highway and Traffic Safety Administration (NHSTA).

The National Fire Protection Association published a series of standards for emergency responses such as NFPA 1561, Standard on Emergency Services Incident Management System and NFPA 1670, Standard on Operations and Training for Technical Rescue Incidents[1]. Among those, NFPA 1561 [82] is widely adopted throughout various states by organizations that provide rescue, fire suppression, emergency medical care, special operations and law enforcement. The NFPA 1561 standard describes, on a high level, the essential elements (e.g., system structure and components) of an incident management system. Figure 2-1 depicts the incident management command structure from the incident commander to responding units. The entire system is comprised of multiple report flows and responsible layers. Specific instructions and recommendations for incident management are also available in the standard. For example, seventeen implementations are suggested to be considered by the incident commander during fire fighter rescue operations such as requesting additional resources and assigning of an Advanced Life Support (ALS) or Basic Life Support (BLS) company.

[1] For details of entire NFPA publications, go to http://www.nfpa.org.

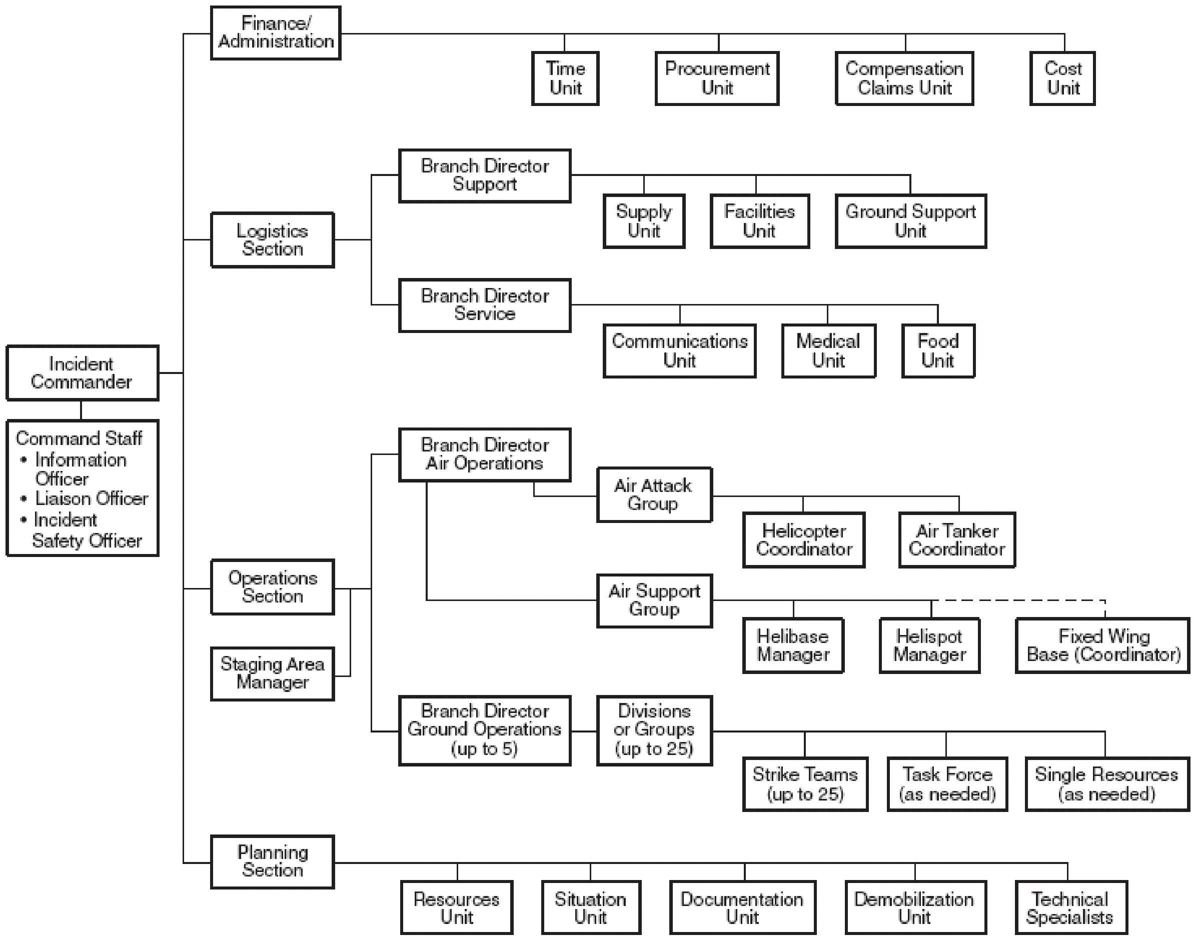

Figure 2-1. Command structure (Source: NFPA 1561 [82])

Based upon the national standards, local governments and agencies establish their operational emergency plans for responding to potential local incidents. The general purpose of such plans is to specifically define task assignments and responsibilities for emergency responding units and personnel in order to best alleviate suffering, save lives and protect property. The Boulder County (Colorado) Office of Emergency Management published a comprehensive emergency operations plan which covers a series of incidents such as hazardous materials incident, tornado, winter storm and weapons of mass destruction [88]. The MODE concept was used to draft the plan. Depending on the number of casualties and the complexity of situation, incidents are categorized into four Modes which follow different response procedures. Mode 1 is least demanding in which the first responders can handle the incident. Mode 4 is most demanding in which mass casualties are involved and State and/or Federal assistance is needed. Modes 2 and 3 events may require regional mutual aid units for assistance.

In the local response plans (see e.g., [85, 88]), checklist, chart and table methods are commonly used to assist decision making, guide command flows and regulate appropriate responses. Figure 2-2 shows a sample of the checklist for medical triage teams. Although response plans provide well-defined instructions, the actual execution is highly dependent on individual experiences because they are just qualitative guidelines rather than quantitative solutions.

E. Triage Teams(s)

_____Obtain briefing from Triage Supervisor.
_____Obtain triage tags from Triage Supervisor.
_____Assess situation.
_____Begin triage of victims.
_____Tag victims with triage tags.
_____Keep Triage Supervisor informed of status and patient numbers.
_____When triage is complete, report to Triage Supervisor for next assignment.

Figure 2-2. Excerpt of Boulder Medical Emergency Response Plan [88]

Klein et al. pioneered research of naturalistic decision making (NDM) in the 1980s. NDM was defined, in short, as the way people use their experience to make decisions in field settings [130]. It mainly studies human decision making in demanding situations such as under time pressure, uncertainty and high risks. In some sense, the naturalistic decision making field is rooted in the military applications of Command and Control (C^2) in hope to better understand human dimensions and investigate the underlying cognitive processes. Klein [61] described two processes used by experienced decision makers to make decisions: (1) using situation assessment to generate a plausible course of action, and (2) using mental simulation to evaluate that course of action. However, the two processes may not be properly implemented due to many cognitive and ergonomic constraints. Purely experience-based decisions are sometimes unreliable and misleading, especially under extreme conditions and unfamiliar, ad-hoc environments.

Masri and Moore [76] defined the context, design requirements and prospects for computer-based, integrated planning information systems. The proposed framework integrates relevant information, knowledge, theory, methods and technology. The information systems used by planners are classified into Simple Systems, Database Management Systems, Decision Support Systems, Planning and Control Systems, Geographic Information Systems, Expert Systems and Integrated Planning Systems. Later, Masri et al. [77] developed an integrated Disaster Policy Analysis System (DPAS) in accordance with the design requirements proposed in [76] to evaluate the costs and benefits of Los Angeles earthquake damage mitigation strategies. The DPAS demonstrated several advantages over former planning systems.

Researchers have investigated emergency response issues in more quantitative and scientific ways including analytical and simulation-based methods. The related literature is reviewed in detail in the next two sections. Goldberg [48] provides an excellent summary of such models.

2.2 ANALYTICAL MODELING

Mathematical programming is a major technique used in operations research to optimize one or a set of objectives under certain constraints. The generic problems can be formulated as follows:

$$\begin{aligned}
&\max_{x \in \Theta} \mu(x) \\
&\text{subject to} \\
&f_j(x) \le b_j \qquad j = 1,2,\ldots,p \\
&l_i \le x_i \le u_i \quad\ i = 1,2,\ldots,q
\end{aligned} \tag{2-1}$$

More specifically, this program seeks to set a variable vector x within the solution space Θ, defined by the constraints such that the objective μ can be maximized. When all the explicit, deterministic functions μ and f_j's only involve linear terms with respect to x which are all continuous variables, the program falls into the category of linear programming (LP). LP has the simplest form and solution scheme in all math programming problems. If some of the variables are discrete (e.g., integers), the formulation becomes a mixed-integer program (MIP). Integer programming (IP) is an extreme case of MIP where all the variables are restricted to integers. Binary integer programming (BIP) is a special case of IP where all the variables are restricted to 0-1 binary numbers. In contrast to the efficiently solvable LP, general integer programs (e.g., MIP, IP and BIP) are NP-hard because a large number of solutions in the feasible solution space have to be explored in order to reach optimality. Nonlinear formulations of the objective and constraint functions make an optimization problem even harder to solve. For such NP-hard problems, heuristics can be utilized to explore only a portion of the solution space and obtain near-optimal solutions in a relatively efficient manner. Some widely adopted heuristic algorithms for deterministic optimization problems are reviewed later in subsection 2.4.1. The details of other solution techniques and their underlying theories for mathematical programming are not the concern of this dissertation, but they can be found in [83] and [118].

In general, analytical optimization models are relatively cheap and faster to solve with proper solution techniques compared to simulation models. Due to this advantage, they are applied widely in many large-scale practical engineering situations although it is sometimes challenging to validate the embedded assumptions.

Mathematical programming methods have been used to solve emergency/disaster management problems. Sacco et al. [99] used linear programming to model the resource-constrained triage decisions for emergency responses. As other resource allocation problems, the objective is to be optimized (e.g., maximizing expected survivors) within time and resource constraints. Yi and Ozdamar [128] presented a two-stage mixed integer program to seek the detailed fleet logistical solutions in response to emergencies and natural disasters such as dispatching commodities, evacuating and transferring wounded victims. The program models the complex problem as a network flow that involves multiple commodities. In the first stage of the program, the emergency vehicles are treated and solved as integer commodity flows rather than binary variables in order to reduce the model size and enhance solvability. Later in the second stage, a simple routing algorithm and a set of linear equations are applied based on the first stage's results to solve for the detailed vehicle fleet operations. Because of the two-stage implementation, the model outperforms other classical formulations in terms of computation time and the outputs can be directly deployed as the dispatching and routing commands. However, the model is relatively inflexible because it incorporates numerous deterministic parameters and coefficients. The author claimed that the program can be flexible and dynamic due to the frequent information updates but gathering the information for the parameters is time-consuming. Furthermore, only the emergency vehicles and commodities are included in the model. The behavior of other major entities (e.g., affected people) in the disaster response system is overlooked.

Queueing methods are another cluster of approaches that have been applied to study dynamic operational systems. Larson [63] first developed an analytical model called Hypercube Model, to study the problems of emergency vehicle base locations and response district design, considering both interdistrict and intradistrict responses. The entire response system is modeled as an expanded, spatially distributed, multi-distinguishable-server queueing system. Each emergency vehicle is described as either free (0) or busy (1) at a time. The system state is regulated by combining each individual vehicle's state. If more than three vehicles are involved, the state space is hypercube in geometry theory from which the model was named. Based on the system state, idle vehicles operate according to the embedded dispatching policies. Since 1974 when the Hypercube Model was invented, many variations have been proposed and implemented to improve the system. Larson [64] modified his original work to develop an approximate Hypercube model called A-Hypercube. The new algorithm is more computationally efficient than the original one and also relaxes the independence assumption for vehicle busy probabilities. Because the analytical formulations are extraordinarily complex, solution algorithms were also proposed by Larson [63, 64]. The hypercube model can be utilized to compute several point-specific and area-specific performance measures such as the busy probability of vehicles. The model itself is descriptive so it cannot improve the solutions unless embedded in optimization routines [48].

In the literature, there exist a significant number of successful applications of the Hypercube model studying the problems of emergency base locations and responses. Mendonca and Morabito [78] applied the Hypercube model to analyze EMS ambulance performance on a Brazilian highway connecting the cities of Sao Paulo and Rio de Janeiro. The major performance measure is the mean response time to emergency calls. The comparisons of original and modified systems were presented to demonstrate that the Hypercube model is effective in addressing the problems of mean response time and workload balancing among the ambulances. Takeda et al. [114] used the Hypercube model to assess the effects of decentralizing ambulances for the urban Emergency Medical Service of Campinas in Brazil. The decentralization strategy was shown, by the Hypercube results, to achieve better performance such as shorter mean response times.

Although the Hypercube model has been predominantly successful in the last three decades, it relies on many assumptions in nature which cannot be alleviated easily. As a major example, the model assumes the emergency calls arrive in the system based on a Poisson process and the service times exponentially distribute with a certain mean – M/M/N queues. The Hypercube model was developed based on Markov models in which the future state is independent of the past states – a memoryless property. Along with many other assumptions, the application domain of the Hypercube model has been tightly constrained. In flexible situations and complex problems, using the analytical model with certain assumptions may be flawed.

2.3 SIMULATION MODELING

Computer simulation is defined by Kelton et al. [58] as "the methods for studying a wide variety of models of real-world systems by numerical evaluation using software designed to imitate the system's operations or characteristics, often over time." Simulation is a popular, versatile and powerful tool because it is capable of realistically modeling considerably complicated and dynamic operational systems. This section reviews several key simulation models and modeling approaches existing in the literature, for both discrete-event and agent-based simulations.

2.3.1 Discrete event simulation

Discrete event simulation models the discrete processes in which changes of the system states occur at isolated points of time [4, 58].

Shuman et al. [102, 103, 104] developed a discrete event simulator in the 80s, called RURALSIM, for designing and evaluating rural EMS systems. RURALSIM incorporates the information of population and geographic characteristics in order to generate multi-type and multi-severity distributed emergency incidents. The responses to those randomly generated incidents are a series of actions: dispatching, field treatment, transportation from field and definitive treatment (when necessary). All of the actions are regulated by a set of operational rules. The details can be found in [102]. Furthermore, the communications between various players, vehicle relocation issues, personnel and equipment configurations are modeled to introduce more realism into the simulator. A number of measures of effectiveness are output from RURALSIM including ambulance response time, satisfactory response percentage, vehicle utilization and so forth. Multiple measures provide decision makers more insights into the system behavior. Several successful implementations of RURALSIM were reported in the states of Maine, Missouri, Oklahoma and Nebraska, respectively [104].

About the same time, Goldberg et al. [47] built a comprehensive simulation model for evaluating a set of emergency vehicle base locations in Tucson, AZ. The model basically simulates a multi-server queueing system in a discrete event fashion and was coded in PASCAL with significant simplifications and assumptions. In the model, the emergency calls are responded by the closest idle vehicle on a first-come-first-served basis regardless of priority or differentiation. The entire area of interest is divided into zones and the travel time (response time) is estimated by the base-zone distance. Because the travel time generation is the only random component in the simulation, a large portion of the paper focuses on modeling travel times by regression using available response time data. The model was validated extensively against the actual data and operations but it is shown that the zone structure is crucial to gaining valid simulation results. This fact restricts the flexible applications of the simulation model. The broken-zone approach can simplify the model and reduce computation difficulty but sometimes it is problem-dependent. Goldberg's model was also applied to answer some important response planning questions such as the effect of adding a resource (e.g., responding vehicle) but the model was built essentially to evaluate the emergency system performance not to optimize the set of base locations. Goldberg [48] provided a summary of the computer simulation models for evaluating EMS services.

2.3.2 Agent-based simulation

A computer agent is an autonomously controlled entity that can perceive its own operations as well as the surrounding environment, compile the predefined rules to make operational decisions, and act based on these decisions. An agent-based simulation model contains a collection of such autonomous decision-making agents and it is preferable in simulating the actions and interactions of the individuals in a network which can affect the entire system [9].

Carley et al. [12] built a multiagent simulation model of bioattacks called BioWar. The model is capable of simulating the outcomes of biological and chemical attacks by building individuals as agents who react and interact with each other in social, health and professional networks. The model incorporates submodels including agent-level disease, diagnosis, treatment, social networks, environmental and attack models. BioWar has been validated against empirical data on different aspects and is capable of evaluating the efficacy of response policies in different areas. As reported, all of the BioWar computations were performed on the NSF Terascale Computing System at the Pittsburgh Supercomputer Center which indicates the model runs require intensive computational power.

Agent-based simulation is a rising area of research and development because of its outstanding capability of capturing system dynamics and interactions. Bonabeau [9] listed four areas of agent-based simulation applications: flow simulation (e.g., evacuation, traffic), market simulation (e.g., stock markets), organizational simulation (e.g., operational risks) and diffusion simulation (e.g., diffusion of innovation and adoption dynamics). It was pointed out that agent-based modeling becomes useful when space is crucial and the agents positions are dynamic, e.g., fire escape.

2.3.3 Hybrid simulation

Recently, a considerable interest in the research and development of hybrid systems has arisen in many engineering and science disciplines. Hybrid systems are complex systems that exhibit both discrete events described by temporal logic and if-then rules, and continuous time dynamics governed by differential and difference equations as well as their interactions [13]. Agent-based simulations (ABS) are a type of continuous systems because the agents' and environmental status are updated much more frequently than discrete event simulations (DES). A hybrid system can be described as a continuous system with phased operation controlled by discrete logic and coordinating processes.

The combination of ABS and DES is a relatively new research area in both academia and industry. It presents an efficient, effective modeling framework in which advantages from both approaches can be utilized. Sridhar et al. [108] proposed a framework for combining agent-based architecture, discrete event system and soft computing (i.e., computational intelligence) methods on one integrated platform called Virtual Laboratory. A fuzzy logic controller was implemented under the Discrete Event System Specification (DEVS). The fuzzy logic rule-based system contains sets of fuzzy "IF-THEN" rules and an inference engine. The rules define the actions and interactions of agents. Lee et al. [68] described an object-oriented approach to model complex agent-based systems. With this approach, the agents of the same type are represented by a class which can interact with other classes of objects more efficiently. A strategic updating scheme is critical for combing ABS and DES because different timing mechanisms are involved. ABS updates the system status in a large number of very small time steps; DES has two principle mechanisms for advancing the simulations: next-event time advance and fixed-increment time advance [65]. In order to synchronize the two systems and maintain the overall computational efficiency, it is desirable to capture the updates and interactions when and only when they occur. Lee et al. [68] used the hybrid simulation to analyze the national airspace system and demonstrated the method's accuracy and efficiency. Gambardella et al. [35] developed an agent-based planning and discrete event simulation system for combined rail/road transport. The agent-based planner is responsible for the dispatching of intermodal transport units from origins to destinations. The discrete event simulation models road transport, rail transport and terminal operations to assess the performance of those plans provided by the former planner. The intermodal terminals are regarded as nodes in a connected supply chain network. The hybrid system has proven to be an excellent means to manage intermodal terminals with pervasive support of information technology. Although it is a type of hybrid systems, the framework constructs the agent-based and discrete-event models apart in two separate subsystems and their interactions are not considered.

With recent advances of computer technology and computing power, a lot of innovative efforts have been made on computer-based hybrid dynamic systems in order to obtain real-time, dynamic and realistic models for large-scale stochastic operational systems. Here, the term "hybrid dynamic systems," distinguishable from "hybrid systems," refers to the computer systems that seamlessly integrate several components/modules, not restricted to agent-based and/or discrete event models.

RealOpt®, developed by GeorgiaTech and some other emergency agencies, is such a simulation and decision support system for large-scale dispensing issues of emergency responses to bioterrorism and infectious-disease outbreak [67]. Formulated as a resource allocation problem, the system focuses on dynamically optimizing dispensing clinics' facility layout and staffing designs to respond to large-scale emergencies. It is coupled with three core components: simulation, optimization, user interface and a linker module, allowing users to enter input parameters dynamically. For the purpose of fast optimization and decision making, a hybrid heuristic algorithm was implemented in the system, combining a greedy algorithm and local search. An actual drill study conducted at DeKalb County, Georgia has shown that RealOpt can provide better designs than the current plans in terms of various measures. Even without direct historical data, which are rare for potential large-scale disaster scenarios, the system still can aid emergency personnel in planning ahead and allocating resources properly by performing "virtual field exercises."

Liu et al. [71, 72] developed an integrated optimization system for planning emergency evacuations. The system combines an analytical optimization model and a microscopic simulation model. The optimization model is a revised version of the cell transmission formulation for network flows and it can efficiently identify effective control plans. The optimized initial solutions are evaluated by the simulation model and presented to responsible system users through an output interface for final decisions. The analytical optimization model is important during real-time operations because it can be sufficiently fast solved compared with the time-consuming simulation model. The embedded microscopic simulation models the detailed traffic conditions and patterns during disaster evacuations which is highly intractable for large-scale, dense networks. The candidate solutions given by the optimization module are just approximately "best" solutions without the considerations of many stochastic factors. However, no further simulation-based optimization procedure is presented in their papers. Besides optimization and simulation, a database module is also integrated in the system to store extensive prior scenarios that have been studied. When similar events occur in the future, the system users can obtain the existing results from the database promptly in stead of running the expensive models again. Some data mining and artificial intelligence techniques must have been applied to implement such a knowledge database system but its details are not described in the papers.

Besides emergency responses, simulation-based hybrid dynamic systems are widely adopted in many other areas such as manufacturing systems. For example, Son et al. [107] constructed an automatic simulation model generator for shop floor control based on a resource model, a shop level execution model and a commercial finite capacity scheduler. The integrated system layout advances simulation modeling by making it simpler and more cost-effective.

2.3.4 Simulation validation

A critical process of simulation studies is validation. It verifies that the model can represent the real system and give realistic results for making reliable decisions. This step normally happens after the model and/or its components are implemented and verified (debugged) with respect to computer codes. It is challenging to validate a complex, large-scale simulation system due to its randomness and numerous internal operations and interactions so the simulation validation itself is an active research topic. It was suggested by several researchers [32, 37, 47, 49] that a single measure might not be sufficient to verify a simulation's model validity so multiple approaches have been proposed.

Gass [37] summarized various validation methods such as:

- Face validation (expert opinion). Ask users and experts to review the model and judge if it satisfies with their knowledge.
- Sensitivity analysis. Investigate how the model behaves when its variables and parameters change and compare to the real-world system.
- Replicative validation. See if the simulation model matches data obtained from the real system.
- Structural validation. See if the model operates in the similar way as the real system to produce comparable behaviors.
- Technical validation. Identify the model assumptions and see if they are close to the reality.

Finlay and Wilson [32] mentioned three methods for managers to perform validation:

- "Validate" the modeler (who builds simulation models) and then trust any models that he/she builds.
- Validate by comparing the internal operations of the model with real systems.
- Treat the model as a "black box" and validate by studying input and output statistics.

Goldberg et al. [47] and Green and Kolesar [49] applied the above stated simulation validation methods to address real simulation studies, respectively and demonstrated the usefulness of those approaches when they are used in combination.

2.4 SIMULATION OPTIMIZATION

Combinatorial optimization problems are concerned with the efficient allocation of limited resources to meet desired objectives when the values of some or all of the variables are restricted to be integral. For example, most airlines need to determine crew schedules which minimize the total operational costs. Constraints on basic resources, such as labor, supplies, or capital restrict the possible alternatives that are considered feasible. Still, in most problems, many possible alternatives need to be considered and one overall goal determines which of these alternatives is best. In the disaster response context, given a region that has a complex transportation network and its emergency resources/assets are dispersed, we want to find an optimal (or good enough) plan for dispatching the necessary equipment to the scene and treating and evacuating the casualties effectively during a disaster. Resources reside in different facilities, having different capacities and operational costs. Due to the stochastic and complex nature of the problem, we prefer to use simulation to evaluate the system performance based upon different criteria, within a mathematical programming framework.

2.4.1 Heuristic and metaheuristic search

A natural approach to solve computationally hard problems is by heuristics. The term was originated from the Greek which means "to find." A heuristic is defined by Reeves [95] as "a technique which seeks good (i.e., near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is." As the basic framework, a heuristic search starts by evaluating an initial solution and moves the search from the current solution to the next one within the solution space iteratively, preferably along a function improving path until a satisfactory solution is identified. Random Search (RS) is the simplest heuristic scheme. It generates new solutions randomly from a predefined distribution to search [129]. Pure RS is only a conceptual algorithm and in practice it is always used with some variations to incorporate useful information from the problem itself or previous searching steps to guide the moving more effectively. Gradient Search (GS) is an improved procedure of pure RS. It moves locally in the most promising path based on the gradient. Due to the utilization of the gradient, GS is generally designed for continuous parameter optimization. The partial derivatives of the objective function $H(X)$ can be calculated to estimate the gradient at a point by $\partial H(X)/\partial X_i = \left(H\left(X_1,\ldots,X_i+\Delta X_i,\ldots,X_p\right) - H\left(X_1,\ldots,X_i,\ldots,X_p\right)\right)/\Delta X_i$ [116]. At least p+1 evaluations of the objective function are needed for a problem involving p decision variables. To obtain the globally optimal solutions, GS should initialize from multiple points and proceed for each of the locations.

By combining heuristics, metaheuristic search algorithms are developed for broad classes of problems that do not have problem-specific solution methods or the solution algorithms are hard to implement. According to the original definition, metaheuristics incorporate both local improvement procedures (intensification) and higher level strategies to escape from local optima (diversification) in hope to reach the global optima [45]. Intensification refers to the exploitation of the previous search results while diversification generally refers to the exploration of the search space [43]. Local Search is an example of the intensification process. It is better applied to situations when the problem structure is well known. Diversification becomes more important and effective when the problem has a general structure which cannot be utilized for identifying the promising areas to search. Intensification and diversification are the two critical concepts which largely determine the behavior of a metaheuristic algorithm. A good balance of them is the key element for a successful metaheuristic procedure. Several well-known metaheuristic search algorithms for deterministic optimization problems are briefly reviewed in the following. The deterministic optimization here refers to the class of problems which have fixed objective functions and constraints and the associate parameters do not change stochastically. Those algorithms form the foundation for simulation based optimization in the current literature.

Genetic Algorithm (GA) [53] was developed by John Holland in the early 1970s. The algorithm's roots are in the field of artificial intelligence. It incorporates the concept of biological reproduction. In every search iteration, GA randomly chooses parent (previous) solutions and combines their components to produce offspring [42, 44]. Scatter Search (SS) [38] is another population-based, evolutionary metaheuristic procedure. Like GA, SS generates new solutions in the form of combinations of existing solution elements but their generation methods are different. Instead of randomization, SS creates new points in the generalized forms of linear combination including both convex and non-convex combinations which might provide some information that is not contained in the original reference points [41, 42, 44]. Tabu Search (TS) [39, 40] was combined with and applied to GA and SS to enhance the procedure (see e.g., [42, 44]) originally but later it was adopted for a much more diverse collection of combinatorial problems. TS basically keeps an adaptive memory in order to utilize the search history to guide the solution process. The memory can help the search avoid reinvesting the solutions that have been explored. Simulated Annealing (SA), derived from the Metropolis algorithm [80], was first presented in the *Science* journal in 1983 by Kirkpatrick et al. [60]. The algorithm utilizes the concept of a metal forming process, called annealing, to search locally for large-scale combinatorial optimization problems. The annealing process is to shape solids (e.g., metal) into a preferable structure by heating the material to the melting point first and then cooling and forming. Likewise, SA sets a high “temperature” parameter to search the solution space wildly at first and decreases the “temperature” gradually to converge to the near-optimal solutions. It is interesting to note that SA allows the search to move in worse directions based on randomization in order to avoid being trapped in local optima. Kirkpatrick et al. [60] and Cerny [14] reported the application of SA to traveling salesman problem in the early years followed by numerous other applications. GRASP (Greedy Randomized Adaptive Search Procedures) is an “iterative randomized sampling technique,” attributed to Feo and Resende [31]. A search iteration consists of two phases: (1) construct an initial solution using an adaptive randomized greedy function and (2) apply local search to improve the initial solution. A generic GRASP pseudo code is provided in Figure 2-3.

```
procedure GRASP()
        InputInstance();
        for GRASP stopping criteria not satisfied →
                ConstructGreedyRandomizedSolution(Solution);
                LocalSearch(Solution);
                UpdateSolution(Solution, BestSolutionFound);
        rof;
        return(BestSolutionFound)
end GRASP;
```

Figure 2-3. Generic GRASP pseudo code [31]

The initial solution construction of GRASP is the crux to the entire procedure. A good solution is formed iteratively by adding only one promising solution element at a time. In each construction iteration, candidate solution elements are evaluated and ordered with respect to a greedy function. Then an element is chosen randomly and added to the solution. Thus, the construction phase incorporates heuristic, greedy algorithms and randomization. Since its invention, GRASP has been applied to many hard problems such as routing, transportation, and location selections [45].

Some important metaheuristic search algorithms are reviewed above. They form the basis for the development of most simulation-based optimization procedures. Although they do not constitute an exhaustive list, they convey a basic, central approach of heuristic search, i.e., start from an initial solution and move to improve iteratively until certain stopping criteria are met. The ideas will be borrowed to create our own metaheuristic algorithms later in this dissertation.

2.4.2 Metamodeling methods for expensive function evaluation

In the optimization context, simulation models serve as the objective functions which need to be evaluated many times to obtain optimal or near-optimal solutions. Since running "black-box" simulations are generally much harder than computing explicit analytical functions, it is helpful to address the issues regarding expensive function evaluation.

While metaheuristic search procedures walk along the promising paths with the hope to reach optimality eventually, metamodeling methods are applied to draw the big picture of "black-box" systems. Metamodeling is also referred as Response Surface Methodology (RSM) in the literature. RSM aims at obtaining an approximate functional relationship (i.e., metamodels) between input variables and output measures. After the construction of explicit metamodels, algorithms for deterministic optimization can be applied to solve the model and seek optimum. Linear regression and artificial neural network are the two most common metamodeling techniques.

Linear regression is a statistical method to fit hypothetical models against observed data. The first-order multiple regression models that involve m independent variables can be formulated as follows:

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \varepsilon \qquad (2\text{-}2)$$

where y is the response variable, x_i's are independent variables, β's are the corresponding coefficients which need to be determined by regression, and ε represents the model's stochastic errors. For large-scale complex systems, nonlinearity often presents where multi-order regression models should be considered. As an example, a generic format of second-order polynomial regression models is shown as follows:

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \sum_{j=1}^{n} \sum_{i \geq j}^{m} \beta_{ij} x_i x_j + \varepsilon \tag{2-3}$$

The x_{ij} terms in the above model represent quadratic (when $i = j$) and interaction (when $i \neq j$) effects of the variables. A validated regression model is capable of estimating the output measures (responses) of the system given the input parameters within certain ranges. Detailed implementations of the linear regression method as well as the associated errors can be found in [81].

Artificial neural network (ANN) method originated from a computer science branch – artificial intelligence – and combines the concept of biological neural networks. The ANN constructs have similarities to its biological counterparts but are much simpler. An ANN normally has dozens to hundreds of nodes (neurons) which are arranged in three connected layers, i.e., input, hidden and output layers. Available input and output data are used to train and verify the neural network before it can be used as a validated metamodel for decision making [50, 52].

Experimental design is another approach to obtaining surrogate values in place of original expensive function evaluations. Brekelmans et al. [11] developed a sequential method to solve constrained optimization problems involving expensive function evaluations. The algorithm creates an experimental design in each of the searching iterations. The design points are evaluated with the underlying functions and the encompassed trust region is approximated by linear regression techniques.

2.4.3 Simulation based optimization

Simulation has great advantages in modeling stochastic systems flexibly and realistically. However, it is a descriptive tool which can just evaluate the system performance with known operational decisions rather than making such decisions [4, 58, 65]. When combined with optimization routines, simulation becomes more versatile. Simulation optimization is an active research area because the marriage of simulation and optimization can provide reliable, realistic and well-structured solutions to a wide variety of practical problems. Simulation based optimization (or "simulation optimization" in short) refers to the problem of maximizing or minimizing the performance of a stochastic system in the format of objective values evaluated by the computer simulation models. The simulation optimization problems can be formulated in a mathematical frame as follows [116]:

$$(\max)\min_{x\in\Theta} H(X) \tag{2-4}$$

where $H(X)=E[L(X,\varepsilon)]$ is the performance measures of the problem which are the expected values from the sample performance $L(X,\varepsilon)$. ε represents the stochastic effects in the simulation system. X is a p-vector of controllable factors and Θ is the constraint set on X. When $H(X)$ is a one-dimensional vector, it is a single-objective optimization problem; when $H(X)$'s dimension more than one, the problem turns out to be multi-objective.

In simulation optimization, the simulation models take the place of explicit mathematical formulae, i.e., objective function values are obtained from simulation runs. Since running simulations is often computationally expensive, it is imperative to apply efficient algorithms to optimize decisions using simulation within allowable computation time. It is the central goal for simulation optimization to reach a near-optimal solution by evaluating as few as possible candidate points. Heuristic or metaheuristic algorithms have been applied extensively to solve the class of optimization problems where expensive objective functions are involved. In the current practice, most simulation optimization algorithms originated from metaheuristics for deterministic optimization [33].

Metaheuristic search-based simulation optimization algorithms typically start from a feasible initial solution and then move the search in generally improving directions until the stopping conditions are met. During the searching process, some methods of randomization are usually utilized to escape from local optima in hope to achieve globally optimal solutions eventually. Andradottir [1] introduced simulation optimization techniques for both continuous and discrete parameter scenarios. Random search was a focused approach to discrete parameter simulation optimization while simulated annealing methods have received great attentions recently. Other general search strategies such as genetic algorithms and tabu search are also applied to simulation optimization successfully [113]. On the commercial/industrial side, Rockwell's Arena integrates an optimization package called OptQuest which mainly uses scatter search and tabu search to perform simulation optimization efficiently [58]. In addition to metaheuristic search, metamodel-based simulation optimization methods also appear in the current literature. Since metamodeling procedures normally involve a large amount of "black-box" function evaluations, they are not applied to simulation optimization in their original forms for large-scale models. Lai [62] applied regression metamodeling with both forward and backward variable selection methods to a large-scale sortation system simulation and used the model to optimize system parameters and package loading policies. The regression models have limited capability of representing the complex simulation system. It was pointed out that multicollinearity is a main cause of regression model failure. Unfortunately, the methodology cannot be applied to problems with a large number of decision variables and dependencies. Kilmer [59] developed the Baseline Artificial Neural Network (ANN) Metamodel Approach to approximate discrete event simulations and it was shown to outperform traditional regression methods. Unfortunately, the related optimization issues were not discussed in the work because the ANN model was used as a descriptive tool as computer simulations. Van Beers introduced the Kriging metamodeling ideas in simulation including the methods to design experiments for Kriging interpolation [5]. Several other simulation metamodeling approaches such as spline, radial basis function, spatial correlation, frequency domain metamodels and their applications were reviewed by Barton [3].

As noted, metaheuristics are created by combining heuristics. A number of researchers have attempted to vary and combine the "standard" metaheuristic methods mentioned above and came up with more powerful simulation optimization algorithms to advance the area. Lai [62] integrated statistical regression and local search procedures to create a new method called Surrogate Search and applied it to solve several simulation optimization problems involving a complex sortation system.

Similarly, a procedure combining a global guidance system, a ranking-and-selection procedure, and local improvement was reported by Pichitlamken and Nelson [92, 93]. In detail, the global guidance system is implemented by the Nested Partition (NP) method to identify a set of the globally most promising subregions in the solution space. The ranking-and-selection procedure incorporates a statistical method called Sequential Selection with Memory (SSM). The memory function can help avoid wasting computation time by reusing the previous simulation evaluations. A hill-climbing (HC) scheme is implemented for local improvement. The neighborhood of the promising subregions is searched for improvement. The NP+SSM+HC metaheuristic procedure is proven globally convergent under very mild conditions and it outperforms other classical algorithms including random search and simulated annealing in the empirical computations. Although the method is more effective, it still needs hundreds to thousands of replications to converge to a global optimum according to the experimental results reported in the paper. For large-scale simulation models, this amount of runs is too many to be a feasible alternative.

On the basis of statistics theory, ranking, selection and multiple comparison procedures are also applied to simulation-based optimization but they perform well only on the problems with relatively small sets of discrete solutions. While not being the focus of this research, an excellent survey on those methods can be found in [112].

2.5 SUMMARY

Various subjects and literature related to the dissertation have been reviewed in this chapter. With respect to decision making, both qualitative tools and quantitative models exist in practice and they are complementary to each other.

Emergency managers, for example, have been using protocols, standards, manuals, tables, charts, and/or even expert opinions for more than thirty years to make effective decisions before, during and after disasters. Those qualitative rules were developed based on experience and practices and can provide substantial insights into the problems.

On the other hand, quantitative and computerized models are more precise and reliable for studying large-scale dynamic systems so they are great tools to aid in making timely and high-quality decisions. Since several decades ago, analytical models such as mathematical programs and queueing models have been implemented for emergency planning and logistics. Those analytical models can typically provide quick, insightful solutions to complex problems.

A lot of successful cases have been reported in the last a few decades that utilized discrete event simulation to model large-scale stochastic systems and obtained favorable results. Agent-based simulation is a relatively new area. It replaces traditional entities with autonomous agents. This modeling approach accommodates more natural logic for most operational systems and it can capture more complex, flexible interactions among entities. However, agent-based models typically consume more intensive computational resources than discrete event models. Thus, the combination of the two (called a hybrid system) presents a promising direction for research. The fact that only a little literature has mentioned this cutting-edge topic further validates the needs for developing a modeling framework for hybrid systems.

Simulation by itself is a descriptive tool which can only evaluate system performance but cannot make decisions. Optimization techniques provide simulation with the power to make decisions. Most simulation-based optimization procedures are adopted from heuristics and metaheuristics for deterministic optimization. Since the execution of complex simulation models is considerably time-consuming, it is challenging for simulation optimization to provide timely and well-structured decisions. Efficient and effective metaheuristics tailored to simulation models are clearly needed, especially for real-time decision making situations.

In the next four chapters, we combine various ideas of simulation and optimization to develop a comprehensive decision support system specifically for disaster planning and management.

3.0 DYNAMIC DISCRETE DISASTER DECISION SIMULATION SYSTEM

Disaster response planning and management are drawing increased attention from politicians, researchers and practitioners due to the huge impact that large-scale disaster events might have on their communities. A comprehensive simulation-based decision support system – *Dynamic Discrete Disaster Decision Simulation System* (D^4S^2) – is developed for the purpose of facilitating the response planning beforehand and management in real time if an actual incident were to occur.

3.1 DISASTER RESPONSE SYSTEM

The major actors in a disaster response system are the first responders and secondary responders. The responders can travel along the city/area's network and either respond to known emergency events or patrol/reserve for potential events. Different cities may develop different disaster response policies and protocols but they are similar in general and guided by the federal response plans. In this dissertation, we focus on developing a simulation model for a typical disaster response system and trying to make it flexible and scalable.

3.1.1 Emergency response system operations and interactions

A city or area can be modeled as an operational network. As a classical network, it consists of nodes and arcs which are the intersections and connecting streets/roads, respectively. More than the network structure itself, various entities and objects are involved in the system. The objects can be either static or movable. Static entities such as structures and rivers normally function at fixed nodes or arcs. Movable entities such as people and vehicles can travel along the network or stay at some nodes to perform their tasks. The entities behave according to certain basic rules and interact with each other in the system. For instance, drivers should obey the laws and rules of the road when driving; when a car meets a person, the car should yield to the person as regulated by the rules. Furthermore, the network resides in the surrounding environment which can also affect the entities' activities. For example, heavy snow can retard traffic significantly.

In an emergency response system, when a disaster occurs and is reported, the responders (e.g., police, fire trucks with fire fighters, ambulances and medical responders) are dispatched to the disaster scene or other critical locations to save lives and assets. The scene could be extremely chaotic because of the excessive congestion caused by both the responders and injured or panicky people. When more responders get involved, other areas might also be affected and the traffic could become more congested. The major disaster event might also increase the number of other related emergency incidents and the response resources might become overwhelmed. It is not feasible to model such a stochastic and dynamic system mathematically, but it is possible to simulate it with operational rules and logic. The more accurate the information and rules used, the better the decisions are made.

Most large-scale disasters involve massive casualties (affected people) at different severity levels. Because human lives are invaluable, the interactions between responders and victims are one of the research foci and the responders' efficacy of treating severe victims is an important measure for the emergency response system. The emergency medical services (EMS) personnel are a generic type of responders who are capable of treating and stabilizing victims at the scene and/or transporting them to medical facilities for more definitive treatment. It is valuable to learn the EMS responder's operations first.

In normal situations, EMS ambulances are responsible for responding to the emergency calls (i.e., 911 calls) which have potential emergency medical needs in their designated service areas. The calls may be served on a first-come-first-serve basis with no preemption and are processed by dispatchers, although there may be an effort to do some basic prioritization when resources are limited. Normally the nearest available EMS vehicle is dispatched. When an ambulance is dispatched, it starts traveling from its current location to the scene. On arriving at the scene, the responder assesses the patient's situation and determines the appropriate actions to take. In severe situations, the responder treats and stabilizes the patient and then transports him/her to an appropriate hospital. In less critical situations, the EMS responder may just treat the patient at the scene and leave him/her for further medical care to be delivered by other support responders. In such a way, the primary EMS responders can respond to most critical needs. After appropriate treatment and transportation, the EMS vehicle becomes available and travels back to the base. From that point, it can be dispatched again to respond to another emergency call either while enroute back to the base or after returning to the base. The above EMS operations are a generic, fundamental response plan which is extracted from the federal, state and local standards (e.g., NFPA 1561 [82], Boulder County Medical Emergency Response Plan [88]) and are being executed nationwide. Some variations may be made to fit the special needs in various places. For example, the City of Tucson, Arizona operates EMS on a two-tier basis [47]. Besides EMS units, they also dispatch fire company resources to assist in responding when EMS services become overwhelmed. This approach can help improve the response service quality but it involves other issues such as mutual aid agreement.

During a major disaster event, the other normal emergency calls within the area should also be covered as much as possible. The consideration of normal call coverage was overlooked or neglected in most of the past disaster management research. In this dissertation, it is integrated as a part of the model and the balance between disaster and normal call responses is studied. In case of a disaster, all available EMS units are divided into two groups. One group is designated to deliver medical care to normal emergencies and the other group is designated to respond to the major disaster. For normal incidents, the designated EMS vehicles respond in the same way as described above. For the disaster, EMS responders can either travel to the scene, stay and stabilize victims at the scene or travel back and forth between the scene and hospitals to evacuate victims, depending on management's decisions.

Triage is a technical term used widely in the emergency medicine literature and practice. It is defined as the process of assessing a group of patients' situations and assigning appropriate medical resources for treatment [99]. It is recommended or required in most mass-casualty situations in order to avoid resource waste and manage limited resources better, especially when medical resources become saturated in a large-scale disaster. As the first step of triage, the victims are normally screened by medical assistants and classified into several categories based on their severity levels (see e.g., [66, 84]). A popular triage coding system for trauma events [84] is presented as follows:

- "Black" or expectant – Non-salvageable/dead on arrival (DOA): Victims who are found to be clearly deceased at the scene with no vital signs and/or obviously fatal injuries.

- “Red” or immediate – Life-threatening injury: Victims who have life-threatening injuries or illness but salvageable (such as head injuries, severe burns, severe bleeding, heart-attack, breathing-impaired, internal injuries). They have the first priority for treatment and transportation.
- “Yellow” or delayed – Severe injury. Victims who have potentially serious but not immediately life-threatening injuries (such as fractures).
- “Green” or minimal – Walking/moderate wounded. Victims who are not seriously injured, quickly triaged, and escorted to a staging area out of the scene for further evaluation and transportation.

RURALSIM is a discrete event simulator developed by Shuman et al. [102, 103, 104] in the 80s for designing and evaluating rural EMS systems (see subsection 2.3.1). It classifies the severity of a patient’s condition in a similar way: dead on arrival (DOA), life threat, severe, moderate and minor [105]. Although the triage coding system is universal to some extent, different response systems use different rules for treatment and transportation of casualties. The basic response principle is to stabilize the casualties at the scene and then transport them to medical facilities as soon as possible according to their priorities.

EMS has more complex operations in the disaster response system compared with other responders including fire, police and hazmat. When they are dispatched, they simply travel to the destination and stay there to perform their assigned tasks individually and/or collaboratively. For example, firefighters are trained for basic life support and they can be the first responders to the scene and work as emergency medical technicians to stabilize victims at the scene; hazmat teams might be needed at the scene to deal with the contaminated materials first before other responders can get into the scene.

3.1.2 Preliminary hard-coded simulation model

The first discrete event simulation model was built in Arena for simulating the EMS system in the Pittsburgh, PA area. This preliminary model contains 103 nodes which represent the key intersections in downtown Pittsburgh. The nodes are connected by arcs which represent the major streets and highways. Emergency responders are simulated to move, perform tasks, and interact with victims and other entities within the network. The EMS operations described in subsection 3.1.1 are hard-coded in the simulator. Experiments were conducted to perform the sensitivity analysis of scene clearance time to dispatched EMS quantity. Six locations (node #8, 54, 60, 70, 84 and 101) were chosen as the potential, experimental disaster scenes and they are marked on the Pittsburgh map in Figure 3-1 along with the EMS bases and hospitals. These locations are spread out the area from north to south and west to east so that they can typically represent a variety of situations in the entire city. Suppose initially there are 30 life-threatening, 30 severe and 30 moderate patients at the scene, so 90 victims in total need treatment and evacuation by the responders. Figure 3-2 depicts the relationship of scene clearance time and dispatched EMS quantity for the six locations, respectively.

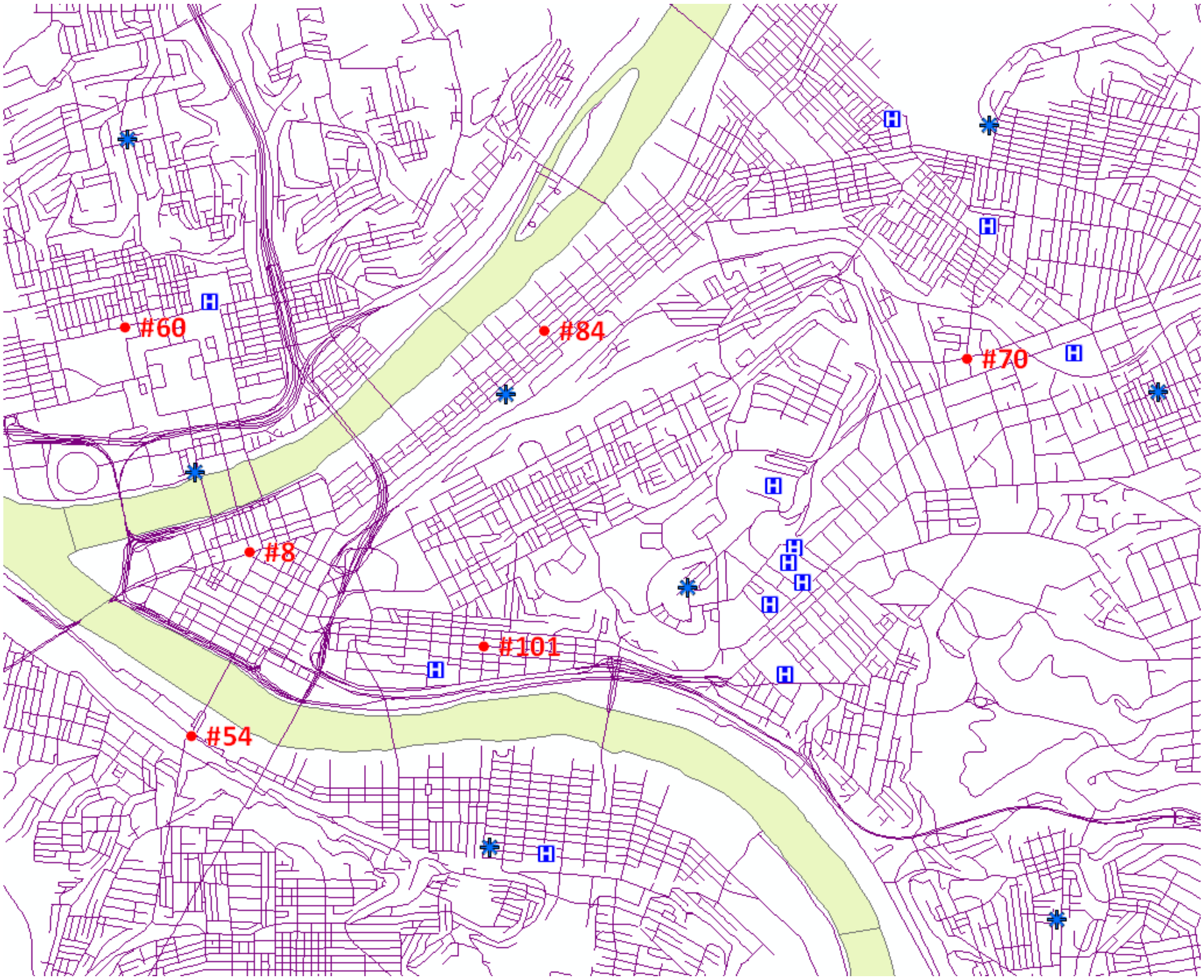

Figure 3-1. GIS map of Pittsburgh EMS bases, hospitals and six experimental disaster scenes

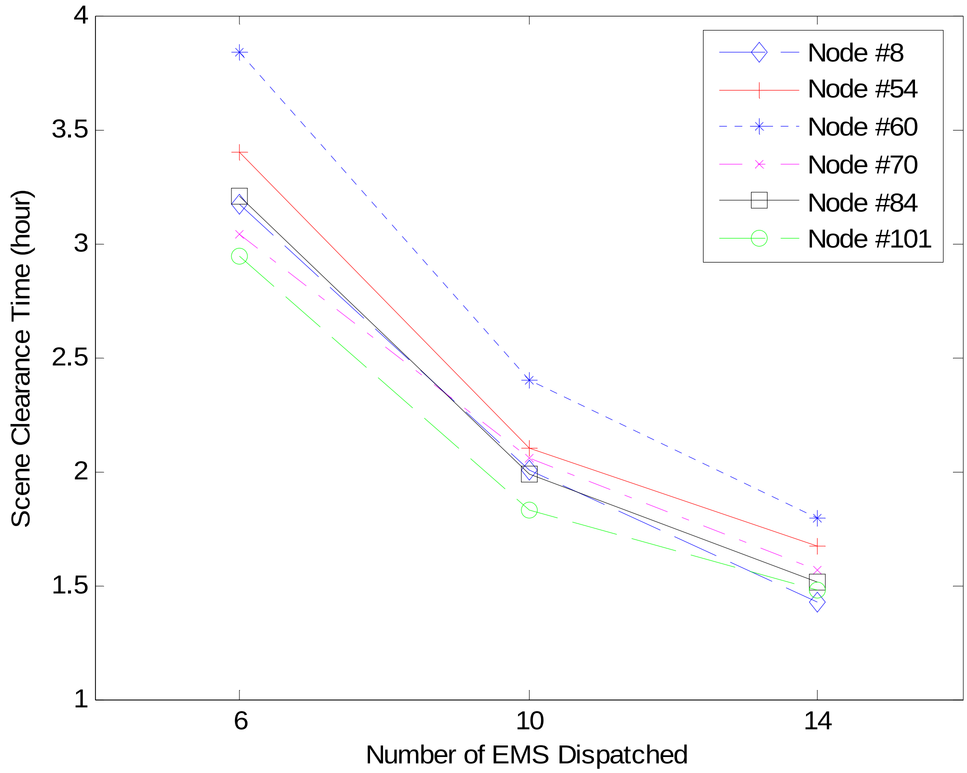

Figure 3-2. Sensitivity of scene clearance time to EMS quantity

It is not surprising to observe that at all of the six disaster locations, more EMS responders (i.e., more responding resources) could help to clear the scene faster. However, the total scene clearance time drops nonlinearly, i.e., more and more slowly, with the linear increase in the number of EMS responders. This phenomenon makes sense because more responding vehicles can cause more congestion at the scene and impact the responders' efficacy of clearing the scene negatively. Another interesting observation is worth noting here. The responses to the disaster scenes #54 and #60 take more time than other locations because they are located northwest and southwest, respectively, to the city and farther away from the majority of hospital resources in the east.

A second experiment was designed and conducted for the sensitivity analyses of scene clearance time and scene fatalities with respect to the number of casualties [123]. It was also used to identify some breakdown points in the current system. David L. Lawrence Convention Center in the Pittsburgh downtown area is a busy location where a lot of traffic passes and complex structures exist. It is a good place to demonstrate the simulator's capability for simulating large-scale disaster events. Suppose a disaster occurs in this location and EMS is dispatched to respond to the event. The structure of victims is 33% life-threatening, 33% severe and the rest ambulatory. The average deterioration rate of life-threatening to death is 2.5% per hour, severe to life-threatening is 2% per hour, and ambulatory to severe is 0.5% per hour. A series of simulations with different casualty scales were run to evaluate the scene clearance time and victim deaths. The results are fitted and depicted in Figure 3-3. The curves show that the breakdown point appears when the number of casualties reaches around 310, after which the number of fatalities at the scene increases exponentially. Thus additional responses are needed to deal with this or above level of events when resources saturate and traffic is highly congested.

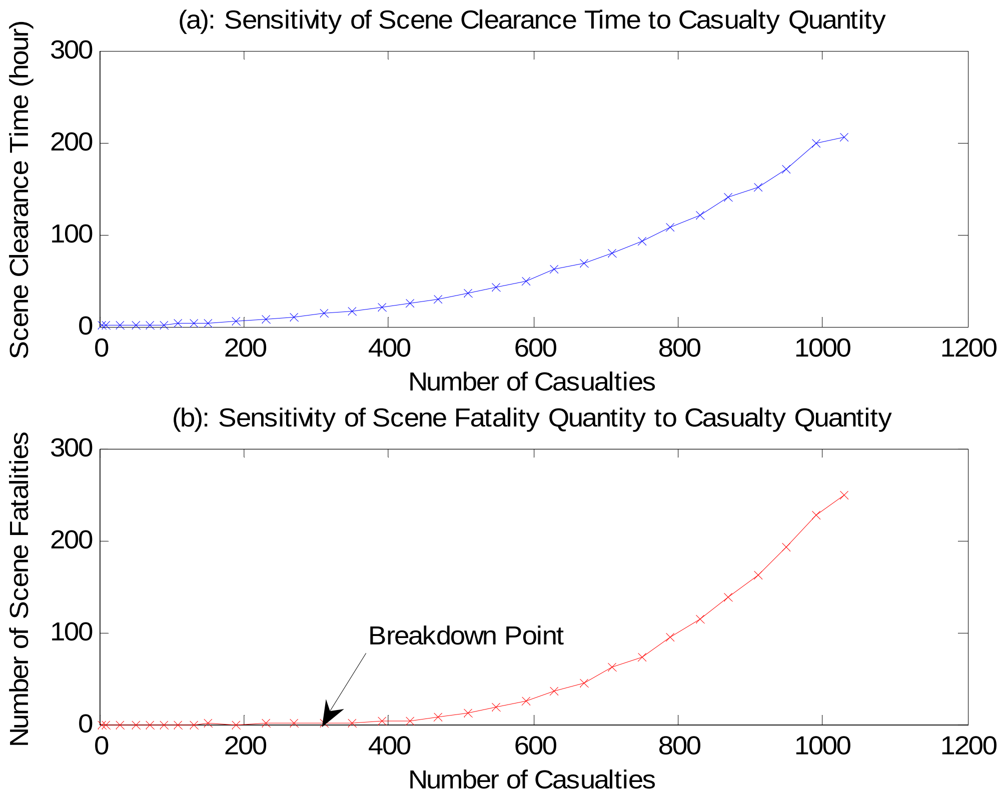

Figure 3-3. Sensitivity analyses for the disaster at the Pittsburgh Convention Center

Although the preliminary simulation model presented above is somewhat effective in evaluating the EMS responses to mid- and large-scale disasters, some major problems prevent us from building a more comprehensive and realistic system, and conducting more extensive experiments. Specifically:

- The simulation model is location dependent: it is a testing model developed for one specific area (Pittsburgh downtown area) and the network with particular nodes and arcs is generated manually. Under this implementation, it is impossible to reconstruct the network quickly if necessary. To make the whole simulation system more portable, flexible and dynamic, an automated simulation model generator using dynamic network data as input is needed.

- Like other traditional simulation models, specific entity operations and logic are hard coded in the preliminary simulator. Because these logical rules are fixed, the simulator has to be remodeled even for minor modifications. Hard coding is not an acceptable approach for flexible system implementations. To make the entity rules more flexible and scalable for modifications and extensions, an efficient simulation-rule interface is needed.
- The current simulator executes for about one hour per run with ten replications on a personal desktop computer with a 3.06 GHz Intel Pentium 4 CPU and 1.00 GB RAM memory. Such a long computation time is not desirable for disaster policy studies or real-time decision making where timely solutions are required. The simulation model is implemented in a commercial available package – Arena by Rockwell Automation. It executes slowly because the classical Arena modeling constructs are used and thousands of modules are created repetitively. The more entity rules that are built in, the more slowly the model runs. To make the model more computationally efficient, a new data structure or simulation scheme is needed.
- The preliminary simulation model is a stand-alone system without any interactions with other component systems. The entities' movement and operations cannot be affected by the change of outside environment/data dynamically. Furthermore, the model only contains a discrete event simulation, and certain continuous submodels (e.g., victim degradation) cannot be handled well. To incorporate required continuous submodels as well as the extensive interactions among entities and components, agent-based modeling ideas have to be introduced into the discrete event simulation.
- Large-scale disasters are low-probability, high-impact events for which the model validation could be extremely hard due to the lack of actual system data. Ethically, we are

not allowed to conduct real disaster experiments for the purpose of validating the simulation. For the computer experiment illustrated in Figure 3-3, domain experts can help examine the results to see whether the curves are consistent with their knowledge and experience in order to at least achieve face validity for the simulation model. However, using the expert experience only does not result in a reliable, solid validation. Several more validation approaches and experiments have to be done to confirm the model's validity.

- Real-time decision making using the validated dynamic simulation system is another important research area. After a major disaster occurs, the emergency managers assemble all types of information and try to make the proper decisions/commands to respond to the event. The simulation system provides the decision makers with a tool to look into the future situations and prepare ahead of time. How to utilize the validated simulation tool to help make timely, dynamic and proper response decisions to the changing situations is a challenge in this dissertation. Simulation-based optimization methods are needed to deal with the mid- and large-scale disaster management issues especially when resources saturate and traffic is highly congested as the breakdown points shown in Figure 3-3.

3.2 INTEGRATED SYSTEM OVERVIEW

The simulation environment is a core component but not the only piece of the disaster decision support system. As an innovation of this research, we interface the discrete event simulation with other interactive modules including a geographic information system (GIS) and real-time information systems to facilitate the synergic decision making process. This integrated simulation system is what we have called *Dynamic Discrete Disaster Decision Simulation System* (D^4S^2).

3.2.1 Work flow

A disaster decision support system is a complex rule-generation system which can assist the decision makers in developing effective schemes to evacuate victims and save lives and property when a disaster occurs. Traditionally when dealing with a disaster, a team of responsible emergency managers will collect the real-time information, assess the situation and issue the controlling commands based on standard protocols, historical data and past experience. However, human experience and intuition are sometimes misleading and could cause the failure of responses because every disaster is unique in terms of its scale, complexity, time and location. During Hurricane Katrina, the Federal government could not assess the situation effectively or get involved in the evacuation based on need, nor could the local organizations. This provides a good evidence of such limitations [2].

In this problem setting, the incident managers need advanced tools to help them predict how to respond to future events and assess the effects of different possible response solutions. They must synthesize a huge amount of information and knowledge in order to develop effective plans. One approach to modeling the system is mathematical programming. Sophisticated mathematical models can be applied to obtain solutions quickly but numerous assumptions must be made to simplify the problem.

Simulation can eliminate many of the assumptions needed by mathematical programming formulations and model the system more realistically. With simulation, we can obtain more accurate results which are critical for informed disaster decisions. Constrained by the computational capacity, the simulation cannot model everything explicitly in the system. One challenge for simulation modeling is how to choose the important entities to model. This involves many tradeoffs which can impact model quality. In the disaster problem, emergency vehicles should be modeled in great detail because their involvement directly determines the system performance, e.g., the clearance time of casualties. In contrast, ordinary vehicles are not decisive elements except that they can create traffic congestion and affect the travel speed of emergency vehicles. A major part of the simulation system involves modeling the road system, emergency resources and entity interactions. The simulation model needs to be calibrated carefully and validated for a wide variety of scenarios before being applied in real-world decision making. A broadly validated simulation model could be more advantageous and precise than mathematical models for complex systems.

Traditionally, simulation is just a system evaluation tool but not for making decisions in real time (or near real time). Our goal is to break its limitations and extend it to be an evolutionary decision driver and optimizer. An evolutionary decision means the decision is not always static after it is made; it can be changed in order to optimize the overall performance as time elapses and the event evolves. Our integrated system works in an iterative way to reason out the proper decisions for disaster management. The system flow chart is depicted in Figure 3-4.

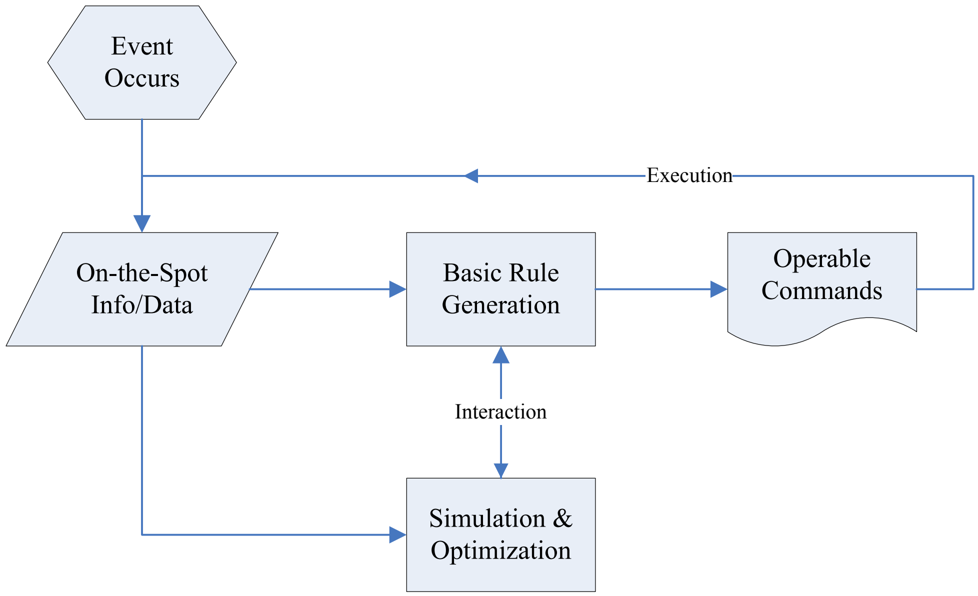

Figure 3-4. Dynamic, rule-driven simulation decision system work flow (adapted from [121])

The diagram above illustrates the basic work flow of the disaster decision support system. When an event occurs, certain information and data (e.g., type of the event, number of victims) are collected from the scene. The data quality affects the system's performance. The more credible the data input, the better the decisions made later. The information/data are then transmitted to the "Basic Rule Generation" and "Simulation and Optimization" modules, respectively. The rule engine can initiate some basic, prompt response solutions to the disaster based upon the initial report of the event and send the response rules to the simulation. The simulator accesses the updated data and operational rules when they are available as the system runs. Conversely, the simulation results feed back to the rule generation module to assist the rule engine in generating better decisions. Mathematical and statistical optimization techniques can also be incorporated into the simulation module to optimize the solutions generated by the rule generation engine. The rule-based system may only include general rules such as "sending Emergency Medical Services (EMS) ambulances to the scene," but it does not specify the optimal rule parameters such as the number of ambulances that should be dispatched to a particular event. In this sense, the optimization can make the general rules more operable by setting the parameters so its results should be included in the plan. The interaction between the rule generation process and the simulation/optimization module is an essential function of the entire decision system. In this system, the simulation is not only a static system evaluator but also a dynamic decision driver. After several iterations, an operable plan will be produced by the rule engine, then justified and sent by the incident commanders, and executed by the emergency personnel to respond to the disaster. A new cycle of the system flow will start by updating the on-the-spot data.

A specific instance can be used to illustrate the working mechanism of the decision making process in detail. Suppose a chemical explosion happened on the corner of AA Avenue and BB Street. Witnesses reported the incident and some descriptive information, saying it appears that approximately 300 people were injured and the traffic in the near blocks was totally congested. The city emergency command center would then be alerted and they would use the D^4S^2 to assist in responding to the event. With the input data from the witnesses, the rule engine generates the first set of rules according to the standard emergency protocols. The rules specify sending all nearby police, five ambulances and two fire trucks to the scene, and close the nearest four blocks after police arrives. The rules are then implemented numerically in the simulation model and the performance is evaluated. The results of the simulation show that it would take 20 hours to clear the scene with the current rules. The information is plugged into the rule engine and/or optimization module which determine that ten more ambulances should be sent to the scene because the long clearance time is not acceptable. As a result, a new set of rules are generated and the system proceeds to the next iteration.

This research focuses on the study of advanced simulation techniques including modeling, optimization and its integration with other interactive modules. We use the simulation as an active decision making tool instead of a passive evaluation tool as is typically done. The objective of building the D^4S^2 system is to help incident managers to rationally design and optimize the responses to various large-scale disasters in hope to enhance the overall effectiveness of emergency responses and reduce the associated risks.

3.2.2 System scope

We develop a dynamic disaster simulation and decision system specifically for simulating and planning large-scale, small-scene, single-major-event disasters. The system is a comprehensive computer-aided planning and training tool available to emergency managers. It provides projected outcomes for various disaster scenarios under different possible plans. The system has three specific usages: planning, training, and real-time decision making/optimization.

There should be no more than one major event presenting in the system at any one time, but a number of small-scale, normal emergency incidents are considered. The major events include the 15 all-hazards planning scenarios for use in national, federal, state and local homeland security preparedness activities, which have been designed by the Homeland Security Council (HSC) [20]. We are particularly interested in studying the abnormal behavior of the whole system when a major event involves many casualties in the scale of hundreds to thousands. The scene vicinity should be relatively compact which means the area can be modeled as a single node within an intra-/inter-connected arc-node network. The response resources should be constrained and conflicting among various necessities.

3.2.3 System framework

As the first step in building a large-scale model, a system framework is carefully designed. D^4S^2 has several module components integrated on one platform for mimicking disaster incidents dynamically and realistically. Figure 3-5 shows the basic framework. Visual Basic (VB) is used as the control structure because a large portion of commercially available software and industrial applications provide VB programming interfaces. For instance, Rockwell Arena for simulation, ESRI ArcGIS for GIS and SQL Server for databases all have such interfaces. This is a tactical consideration for long-term development as the VB-structured system is well scalable to other software and applications.

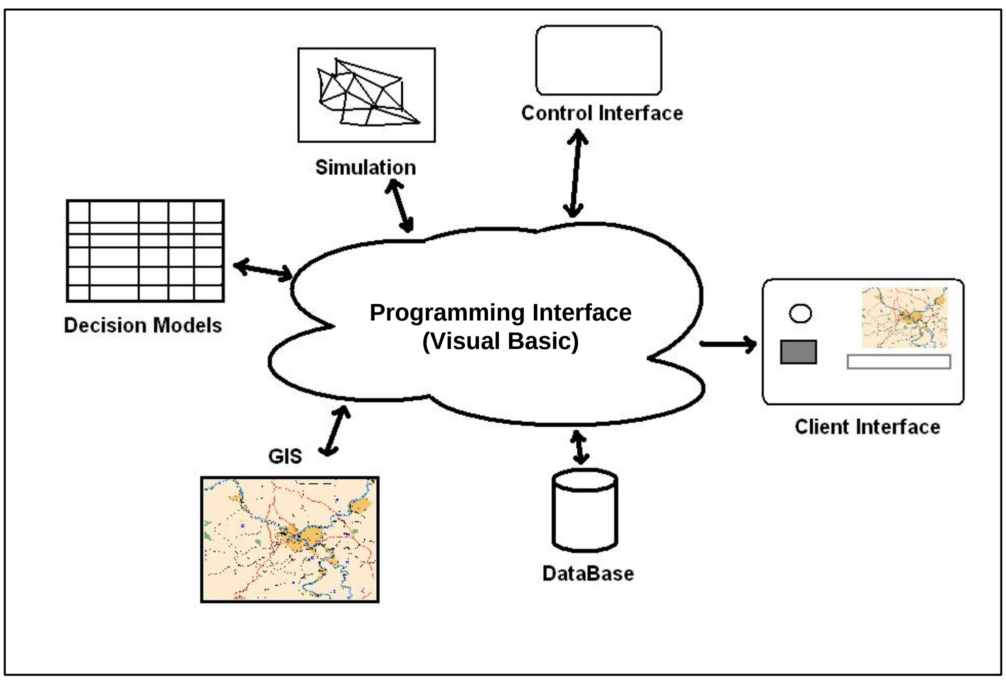

Figure 3-5. D^4S^2 system framework [122, 123]

A simulator, a GIS, a client/control interface, and several decision models are linked together by a relational database to share data, simulate and optimize the disaster responses. The system data flow is depicted in Figure 3-6. The flow mainly consists of three parts: a VB application, an intermediate database and the Arena simulation package. The VB application initially prepares the data needed for running the simulation such as the GIS data, event type and size parameters. The data are stored in a well designed relational database. Arena then retrieves those data and runs several replications. Progressive results are collected iteratively and stacked in the database during the simulation run. Finally, useful results are extracted and compiled by the VB application and displayed on the client interface for view and analyses. Using a database as the system media provides a convenient interface and enhances flexibility for the end users. The users can simply query and update the information in the database to interact with the simulator dynamically whenever better information is available. Some key tables in the database that describe the simulation are summarized in Table 3-1.

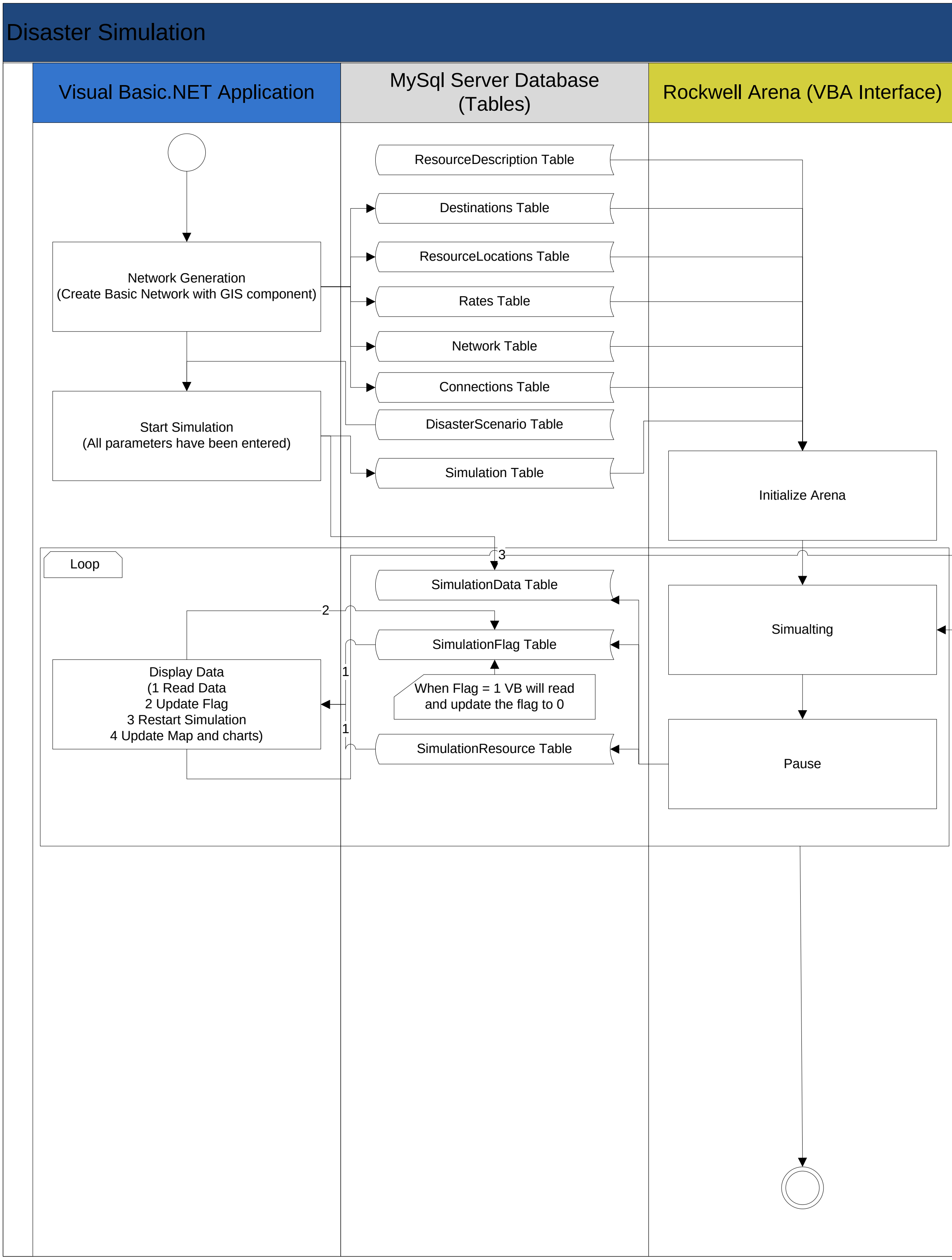

Figure 3-6. D^{4}S^2 internal data flow (adapted from [123])

Table 3-1. Database tables for describing simulation (adapted from [123])

DB Table Name	Contents
ResourceDescription	Descriptions of emergency resources, e.g., EMS ambulances.
Destinations	The destination points of emergency resources, e.g., hospitals.
ResourceLocations	Emergency resource locations, e.g., fire stations.
Rates	Emergency vehicle nominal traversal times on arcs.
Network	Road network data, e.g., node positions.
Connections	Road network connectivity.
DisasterScenario	Disaster type information, e.g., hazard materials.
Simulation	Setting parameters of the simulation model.

The core simulation primarily deals with a complex network flow problem which involves the entities' movement and designated actions defined by a set of rules. It is driven and changed dynamically by other components such as a rule generation engine and a GIS. GIS is defined as "a computer-based system to aid in the collection, maintenance, storage, analysis, output, and distribution of spatial data and information [8]." In our context, the GIS maintains all the important geographic-related metadata which describe the simulated network and relevant attributes such as node positions, arc connections, and assets distributions. Additionally, the GIS provides a collection of spatial analysis tools which can be utilized to perform pre- and post-simulation analyses and assist in making decisions. In such a setting, the system carries its dynamic nature along with randomness since the geographic information can be changed and retrieved in a real-time manner. In addition, the rules that regulate the entities' actions can be revised by either the incident managers or certain threshold conditions encoded in the rule engine. This process complies with reality because disaster management is essentially an ad hoc activity and the decisions should be dynamic with the progress of the event.

Because GIS systems can provide rich geo-metadata (e.g., spatial, resources, weather) for most areas in the world, our disaster simulation system is no longer location dependent as was the first model presented in subsection 3.1.2. The geographic information is stored independently from the simulation. An automated model builder is developed to retrieve the GIS map data and construct the network simulation model automatically. This implementation can give the simulator the greatest flexibility on deployment: as long as the GIS data are available, we can simulate with a short lead time. As stated before, the simulation model is mainly a network problem. When model size and computational efficiency are considered, not all nodes can be included in the model. Strategically, finer grids are modeled for the more interesting areas (e.g., street blocks around the disaster scene) while cruder grids are built for other less interesting locations. With tools like GIS and automated model builder, dynamic network construction "on the fly" during the disaster becomes possible as we can easily and quickly change the network settings. Several researchers have done some work in combining simulation with GIS and/or rule-based systems. Wiley and Keyser [119] incorporated accurate GIS data with transportation simulation models to address the traffic incident management issues. Born [10] utilized a specific discrete event simulation language WebGPSS which is teamed with a GIS system to simulate business operations. Cheng [15] proposed a rule-based simulation model for the train traffic network which uses "IF-THEN" rules to drive the simulation runs. However, the highly integrated, dynamic framework for the network-centric hybrid simulation and decision system presented here is unique.

3.3 SUMMARY

In this chapter, some preliminary work illustrates the drawbacks of a traditional simulation model and demonstrates the need for a more flexible, integrated system for large-scale emergency management. First of all, following a set of basic emergency response rules, a hard-coded, small-scale disaster response simulator is constructed as a testing bed. Several major limitations are reflected by the preliminary results such as location dependency, rule inflexibility, excessive running time and lack of system interactions. To alleviate these limitations, a more flexible, dynamic system – Dynamic Discrete Disaster Decision Simulation System (D^4S^2) – is demanded. This integrated system's work flow, application scope and framework are then described. The next chapter develops the critical concepts and methodologies for combining agent-based modeling and discrete event simulation. An efficient and effective framework is required for use in the real-time decision making situations.

4.0 AGENT-BASED DISCRETE EVENT DISASTER SIMULATION

In this chapter, the core simulation component of D^4S^2 is developed and implemented. The simulator bears the traits of classical discrete event simulation as well as agent-based modeling concepts. Several approaches to integrating agent-based, continuous models into the discrete event simulation framework are discussed. The complete simulation model is validated from various angles.

4.1 MOTIVATION

In discrete event simulation (DES), the system status changes and updates are only driven by events; while in agent-based simulation (ABS), such changes are determined by all the agents' status and environmental situations. Generally speaking, DES is more computationally efficient than ABS because the latter has smaller update intervals and demands more frequent system checks and changes. Although both are large-scale models, Lai [62]'s discrete event simulation can be run on a personal computer within 20 minutes while Carley et al. [12]'s agent-based simulation has to be executed on a super computer. In order to build our simulation system more compactly and efficiently in hope to use it to make timely decisions, we choose the discrete event framework as the core simulation structure. However, the simulation is not the only piece in the disaster decision support system. To facilitate the entities inside the simulator better interacting with each other as well as communicating with the outside environment, the agent-based modeling concept is incorporated. Furthermore, the agent-based implementation is also needed for several continuous submodels in the system.

4.2 MODELING METHODOLOGY

4.2.1 Language and software selection

One crux of this research is the simulation interoperability, i.e., the capability of different components to operate and interact with the simulator seamlessly on one platform. Developing a stand-alone simulation package is not the main focus. Instead of recreating the wheel, we have been urged to choose an appropriate, mature, existing simulation software package. A preferred list of important criteria is the first step to getting started [86]. By analyzing the nature and the integrated structure for the complete simulation decision system, we are able to list some of the most critical criteria for the simulation package selection:

- Large-scale modeling capability. The disaster response system involves a huge network with many rule-driven entities.
- Interoperability with other packages. The ability of seamless integration with other modules (e.g., GIS) is one of the most important factors to be considered in this research. This capability also impacts the future scalability of the system significantly.
- User-friendly interface and debugger for error checking and code tracing. Visualization of the situations is an essential functionality of the integrated system to assist the human

experts in better understanding their problems. Also, the well designed interfacing tools may help the system developers to detect errors in the process of development.

According to an *OR/MS Today*'s survey of dozens of simulation software packages [89], Rockwell Arena® is one of the best options to fit our research, because:

- Arena is developed with the advanced and reliable simulation language SIMAN, which can be used in many application areas. Arena has the most comprehensive modules and processes among existing packages for discrete-event and flow process simulation because it combines most features of many other packages. Compared to other packages, Arena is a simulation package that is particularly good at modeling large, complex systems (see e.g. [62, 107]).
- Visual Basic for Applications (VBA) interface support. Visual Basic (VB) is the most widely adopted language in business and industry because it can be easily interfaced with various application packages. We use VB as the control structure because it has a lot of advantages in system integration. VB's neat interfaces can seamlessly pass data among applications such as Arena, ArcGIS (GIS software) and SQL Server. VB programs are portable to handset devices such as PDAs so it also enables us to equip individual responders easily in the future [120].
- Arena provides a user-friendly graphical user interface (GUI) for building and debugging simulation models. The powerful visualization and animation tools can help the users to identify problems and flaws effectively. Experts who have little knowledge of the simulation techniques can also benefit from the GUI to obtain good insights into the problem and make appropriate decisions.

4.2.2 Agent-based discrete event modeling

The concept of agents has been used in the artificial intelligence (AI) field to model real-life intelligent entities. A computer agent is defined as an entity or object which can be controlled autonomously. The agent is capable of perceiving its own operations as well as the surrounding environment, compiling predefined rules, making operational decisions, and acting based on these decisions [98]. This process is similar to the human's "thinking" process. An agent-based model consists of three key elements: autonomous agents, environment or space, and rules that govern the agents' movement and interactions [28]. It is best for simulating complex, dynamic systems. Based on this rationale, we want to synthesize the agent-based idea with the discrete event simulation to simulate the behavior of responders for various disaster scenarios.

In responding to a disaster, the responders normally utilize special vehicles (e.g., ambulances, fire trucks). Hence every responding emergency vehicle can be regarded as a unit and modeled as a rational agent in the simulation system. Different types of agents many have different attributes and operational rules. For example, an ambulance needs to travel back and forth between the scene and hospitals, transporting patients continuously while a fire truck and the firefighters can stay at the scene to handle special fire situations. The responders are instructed of their actions by the commands or their own judgments authorized by the predefined rules. As an example, the EMS ambulance agent actions and its interactions with other agents are depicted in Figure 4-1.

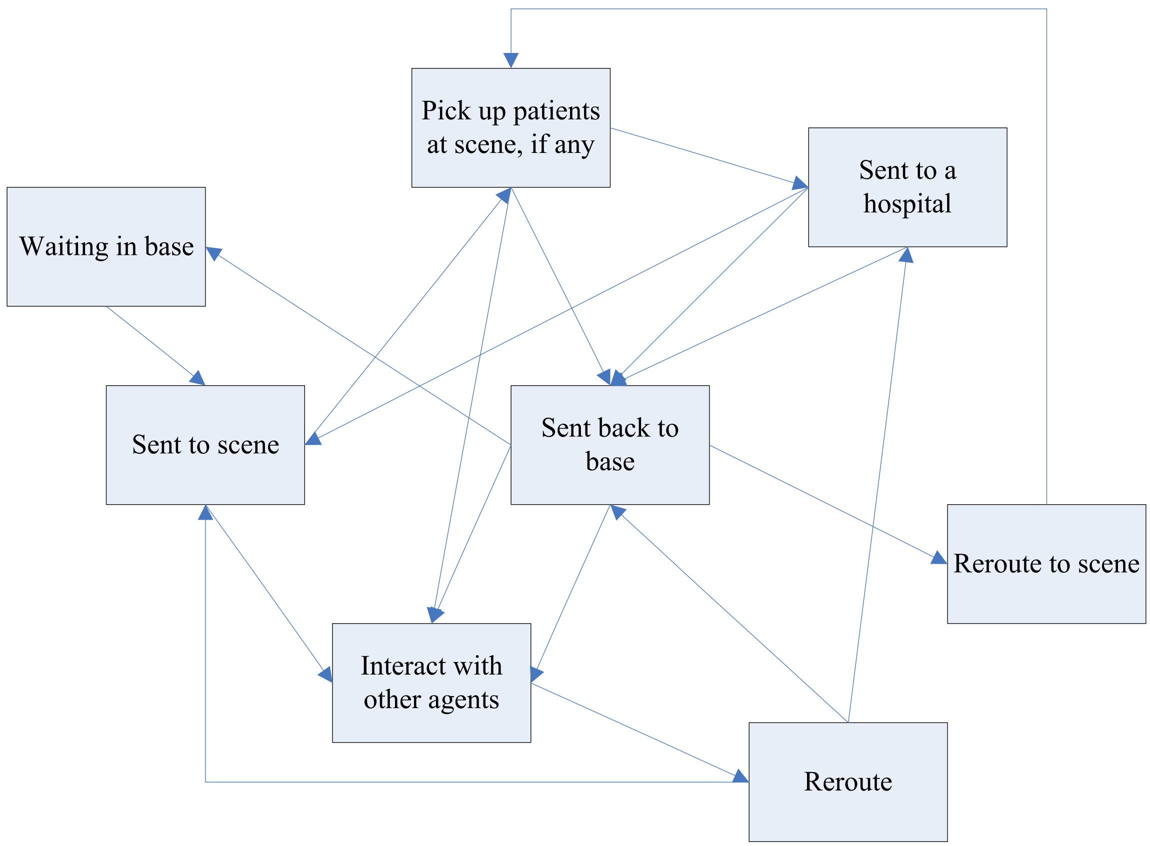

Figure 4-1. Agent actions and interactions of an EMS ambulance [121]

The ambulance agent actions and interactions with other entities are defined by a set of ambulance dispatching and operational rules (e.g., [47, 88]). Other first responder and secondary responder agents and resources, including EMT, paramedic, fire, hazmat, medical helicopters and mutual aid vehicles, are built in the similar manner in the simulation system. The agents are not limited to the responders. Any objects in the system can be modeled as agents if they interact with others and/or the environment.

The agents' attributes are used to define their operational and environmental status. To enable the dynamic status changes, all of the defining attributes should be parameterized to quantitative variables and maintained in the database. Table 4-1 lists the attribute definitions of responder vehicles and streets. It is worth noting that besides the designated responder vehicles, other relevant objects such as emergency assets (e.g., fire hydrants, medical suppliers) and city infrastructure (e.g., streets, bridges) are also tagged by their defining attributes. These static objects (agents) may interact with the responders so they are also important in the simulation model. Some of the attribute values are fixed while many others are variable and updated as the simulated event evolves. The agents' status is critical information for decision makers who observe the system's behavior and develop the proper responding plans and decisions.

Table 4-1. Attribute definitions of sample agents (adapted from [121])

Object	**Attribute**	**Property**	**Variable Type**
Responder Vehicle	Vehicle ID	Fixed, read-only	Integer ID
	Vehicle Type	Fixed, read-only	Integer ID
	Trip Start Node	Dynamic, simulation	Node ID
	Trip End Node	Dynamic, simulation	Node ID
	Current Node	Dynamic, simulation	Node ID
	Next Node	Dynamic, simulation	Node ID
	Last Action	Dynamic, simulation	Encoded integer
	Current Action	Dynamic, simulation	Encoded integer
	Next Action	Dynamic, simulation	Encoded integer
	Action Parameter	Dynamic, simulation	Integer
	Queueing Priority	Fixed, or dynamic	Integer
Street	Street ID	Fixed, read-only	Integer ID
	Connectivity	Fixed, read-only	Node ID
	Lane No	Fixed, read-only	Integer
	Speed Limit	Fixed, read-only	Integer
	Condition	Dynamic, GIS	Encoded integer
	Congestion	Dynamic, GIS, simulation	Floating-point

In the real world, most events and physical models are time-continuous. The agent-based simulation is also time-continuous because the system needs to check and update the agents' and environmental status very frequently (i.e., continuously) to achieve accurate results. This is one main reason why the time-continuous simulation model is so computationally expensive; hence, a pure agent-based simulation model does not meet our requirement of building an efficient decision support system. Compared with the agent-based models, discrete event simulation updates the system only when an event ends and the next event starts. The less frequent update time step makes the discrete event models more efficient. As a major thrust of this research, we synthesize the agent-based and discrete-event simulation in order to maintain both model quality and efficiency. An important and challenging issue in integrating the two is how to keep the integrity of updates among different models, while letting the agents interact with the environment in a consistent and efficient way [68]. In other words, we need to find a way to properly break the time-continuous, agent-based process into separate, discrete events and check/update system status only when necessary. Fortunately, most of the events can be discretized into allowable time fragments and modeled as discrete events under certain reasonable assumptions.

Developing the agent status update scheme and the simulation data structure for network-centric models is a major contribution of this dissertation. In the disaster response system, the main "actors" are the responders. Because the responding personnel normally utilize emergency vehicles and equipment (e.g., EMS ambulances) to respond, such vehicles and equipment can be modeled as agents. In the network-centric model, the moving agents travel along the network and perform their designated tasks on particular nodes. The simulation is expected to check and update its status when any agent (e.g., a responder) finishes one action and starts another action.

In such a way, the complete continuous process can be broken into a series of node-related actions in a discrete-event fashion. When an agent arrives at a node, it may not perform any tasks, perform only one task, or perform multiple tasks. It then leaves the current node and proceeds to the next node. In each network node, only three functions are needed: Delay, Hold and Travel. The "Delay" function can impose a time delay on the agent which simulates its actions or tasks such as loading and unloading patients. We assume if an action is initiated (i.e., an entity enters a delay module), it cannot be ceased in the middle. The "Hold" function can make the agent stay at the current node until released in the future. The "Travel" function determines the next node to which the agent should move. Thus, every individual agent has three mutually exclusive and exhaustively collective states: Tasking, On Hold and Traveling. The states are interchangeable and they change dynamically based on the status of the simulation and predefined rules. The programming flowchart is depicted in Figure 4-2.

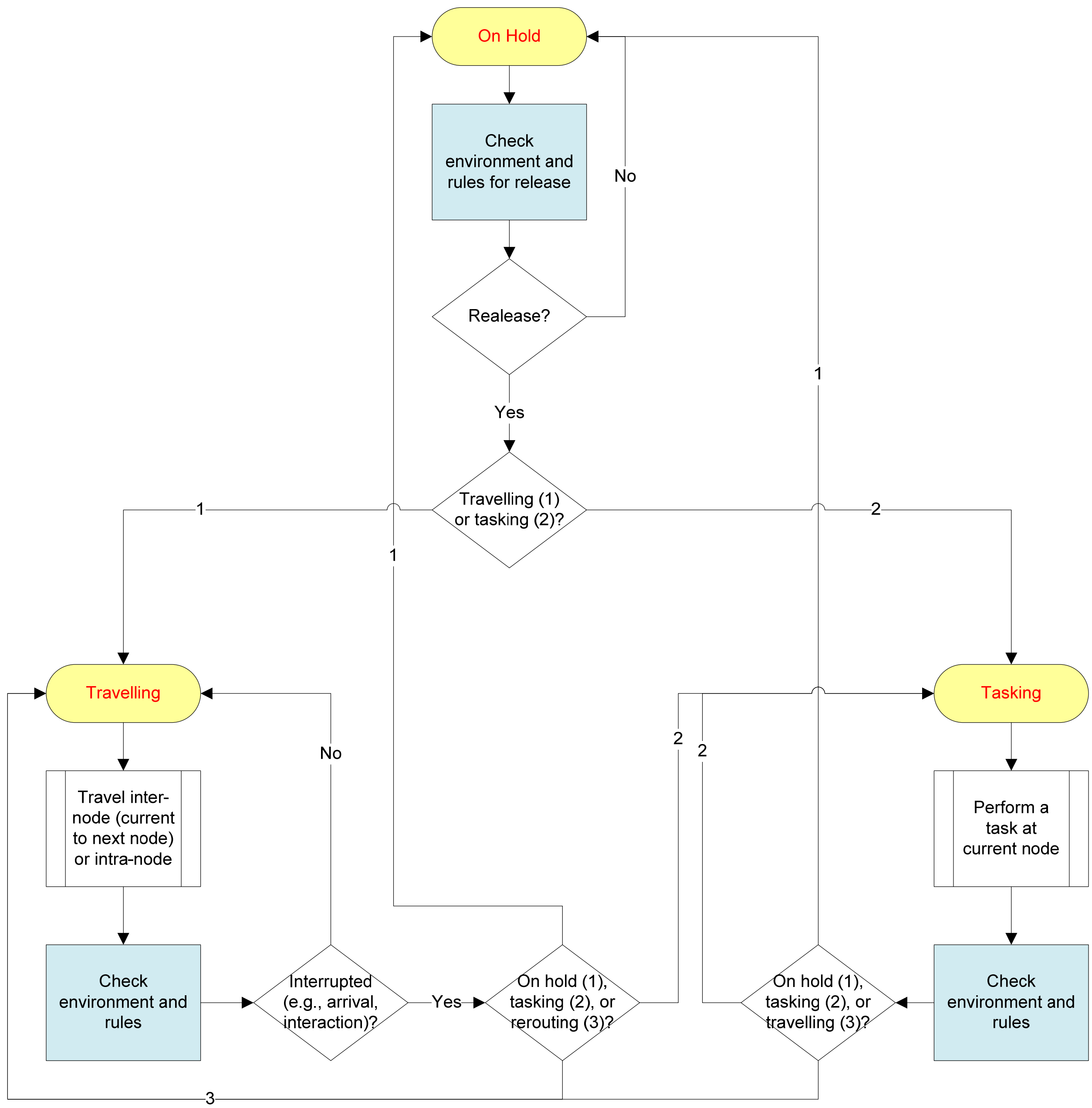

Figure 4-2. Flowchart of moving agents' three states

According to the above rationale, a unified data structure for the network nodes can be developed as follows:

- Every network node is a submodel. All the node submodels are connected by the arcs to form a network.

- Every node submodel has the identical internal structure with only three components: Task Delay, Hold and Travel.
- The different agent-specific actions simulated by the Task Delay modules have to be encoded numerically so that they are tractable by computers. Such encoded numbers can be easily decoded if the information needs to be presented to human users. Some of the EMS ambulance actions are encoded in Table 4-2 as an example. When an agent performs a new task, its action attributes (defined in ***bold italics*** in Table 4-1) should be updated accordingly.

Table 4-2. EMS ambulance agents' action codes (adapted from [121])

Vehicle Agent	Numerical Code	Action/Task Description
N/A	0	Unknown or N/A
EMS Ambulance	500	At base wait for call
	501	At base dispatched and process
	502	Travel from base to scene
	503	Travel from hosp to scene
	504	Pick up victims at the scene
	505	Travel from scene to hospital
	506	Drop off patients at hospital
	507	Travel from scene to base
	508	Back to base and process
	509	Hospital process after drop-off
	510	Travel from hosp back to base

The above unified node submodel has many advantages besides enabling us to incorporate the agent-based concepts. The data structure can help separate the simulation model from the driving agent rules so as to make the rule base more flexible to changes and scalable to extensions. The simulated logic/rules are no longer hard-coded as in the traditional simulation models. The agent rule base is discussed in the next subsection. The unified node data structure also makes the simulation model location independent since the simulated network can be reconstructed quickly when the structure data (e.g., node positions, node connectivity) are changed. An automated simulation model generator is developed for this purpose.

4.2.3 Rule-based System

In the simulator, the responders are modeled as autonomous agents who can "analyze" the environment by themselves and generate actions based upon predefined, specific logic/rules. The agent rules are the key driver of the simulation system because they decide the agents' behaviors and drive the simulation forward. The specific operational rules may vary across different areas. Even for the same area, the rules may change in different situations. As described before, in order to maximize the rules' flexibility and scalability, we maintain the rules outside of the simulation model and break their dependency in the implementation. The collection of rules forms a database, called rule base, and it interacts with other components during the simulation run. Such a system is a rule-based system. The rule-based simulation system can simulate the decision making process of emergency responders and incident commanders. Traditionally, a rule-based system consists of a rule base with permanent data, a workspace or working memory with temporary data, and an inference engine. A user-friendly interface can be a plus to the application but is not critical to the basic reasoning process [55]. The architecture of the ruled-based system integrated with the simulation and GIS is depicted in Figure 4-3.

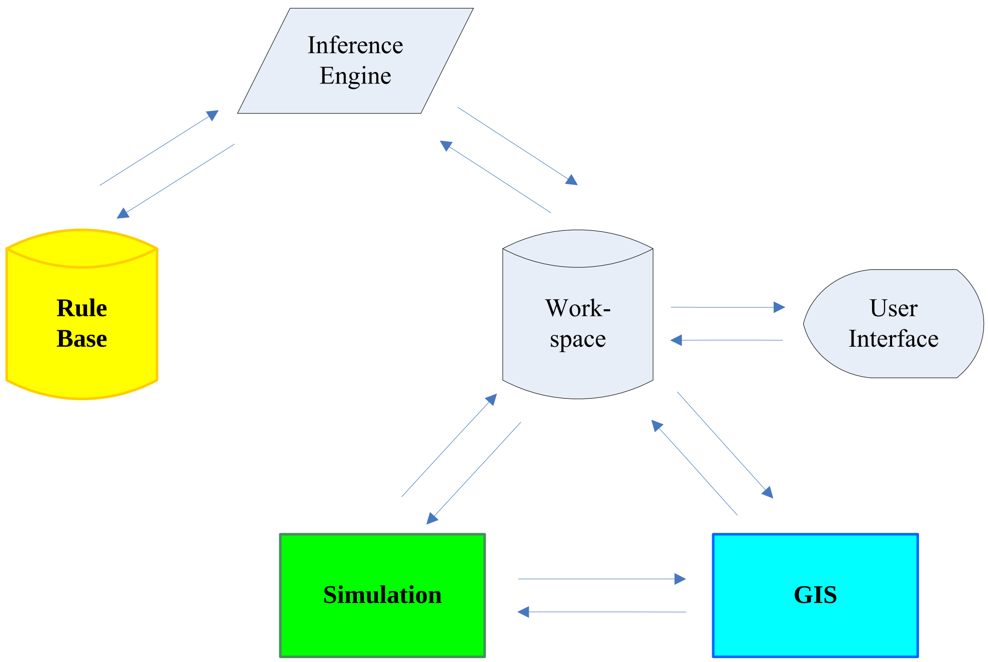

Figure 4-3. Rule-based, integrated system architecture (adapted from [124])

The knowledge used by the responders, incident managers and other decision makers is stored as pieces of rules in the rule base. More precise rules can help the system to generate better results. Rules are typically in the format of "IF-THEN" and extended "IF-THEN-ELSE" clauses [55] as follows:

IF some condition(s) THEN some action(s)

IF some condition(s) THEN some action(s) ELSE some action(s)

The clauses can be expanded by attaching attributes such as the probability of certain consequences if the plan is implemented. The workspace is a collection of databases that store the temporary fact data about the system. The data come from the simulation, rule bases and other integrated applications such as the GIS. The simulator, rule bases, GIS, and other components update the databases "on the fly" as the event evolves. The inference engine determines how to pick and apply appropriate rules to the working memory and execute the rules. The execution of a rule may change the facts in the workspace either immediately or after a period, and those changes could trigger other rules. The user interfaces visualize the evolving situation and the decisions, and also facilitate human decision makers to interact with the system. Enabling the human experts to track the system's progress can help them identify some unrealistic or defective rules and enhance their management experience. Figure 4-4 shows the D^4S^2 user input interface for generating a disaster simulation. Users can specify the disaster parameters such as event occurrence time, event type, and size with victim severity distributions. Users can also operate the GIS system through the interactive interface to manipulate GIS data such as network nodes and connectivity, and resource locations (e.g., hospitals, fire stations). Then, the simulation network model can be created automatically with the necessary GIS data by a computer program. After the simulation run, various resultant charts are displayed on the output interface shown in Figure 4-5. Those charts depict the progressive situations of the event by breaking the results into consecutive segments over time.

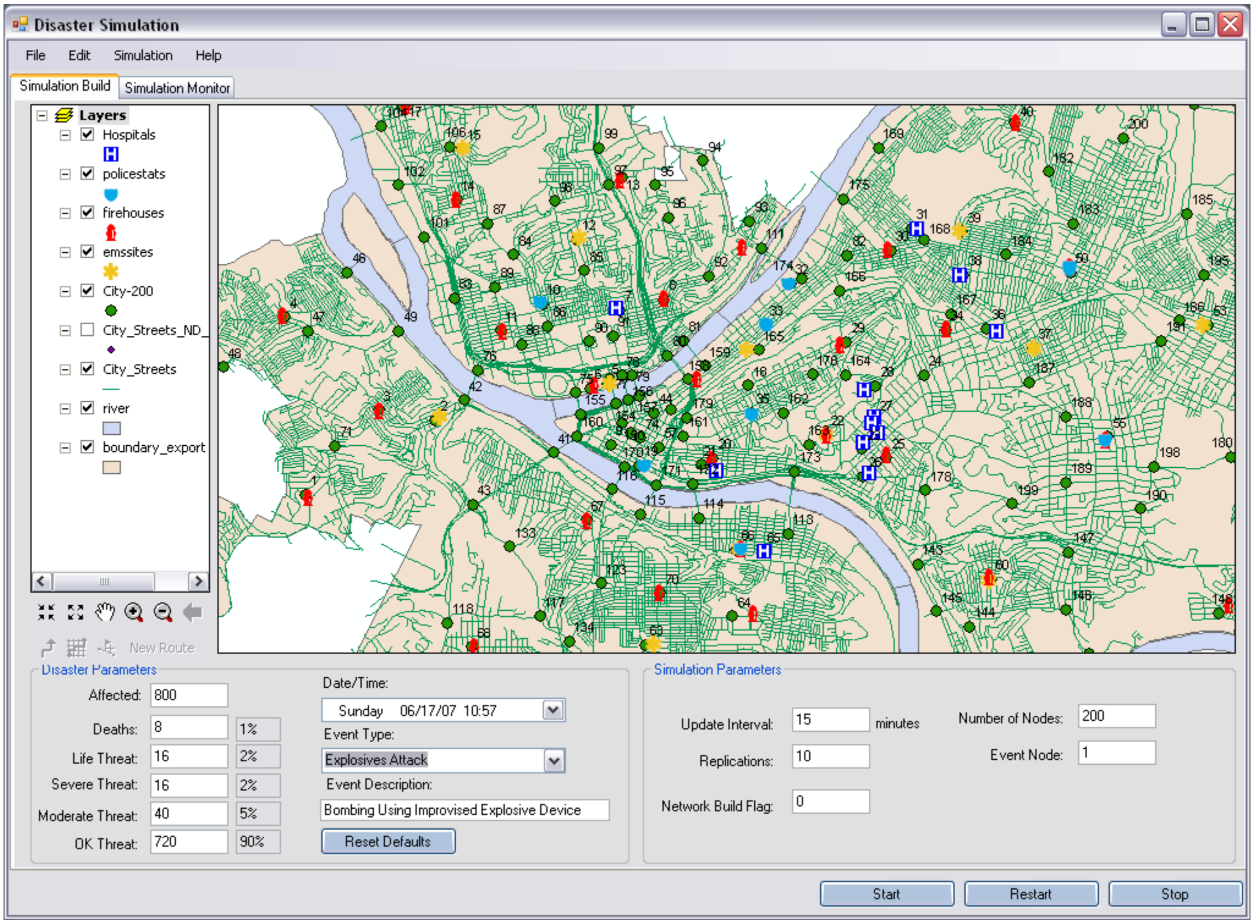

Figure 4-4. D^{4}S^2 GIS and user input interface (adapted from [123, 124])

Figure 4-5. D^{4}S^2 simulation result output interface (adapted from [123, 124])

The rule-based simulation model mimics human thinking processes. When a responder agent finishes one action, it will "think" about what to do next so that the embedded computer programs will be executed to facilitate the "thinking" process, just like a human's brain. The main responders' operational rules are summarized and described below. The rules focus on general trauma injuries and all of them comply with the national response standards.

EMS Personnel and Equipment:

In our system, modeling the emergency medical services (EMS) is one focus and the EMS ambulances have the most complex operations and interactions. During a major disaster event, the EMS ambulances can respond to the disaster or normal emergencies in the area. When responding to the disaster, they can evacuate the victims to hospitals or stay at the scene to treat and stabilize the patients. Normally, an EMS system is equipped with two types of vehicles and personnel: advanced life support (ALS) and basic life support (BLS) units. Some other non-traditional equipment such as medical helicopters may also be utilized to respond to the disaster. These have different medical treatment capability and capacity.

- *If* dispatched, *then* process (assemble driver, medical responders and equipment) immediately.
- *If* dispatched and ready, *then* travel to the scene.

  For disaster stabilization:

- *If* arrive at the scene, *then* stay, treat and stabilize the patients until further instructed. The EMS involvement can improve the patients' survival probability. Such improvement is different for ALS and BLS units.
- *If* finish the current task and no further instructions, *then* travel back to the base.

  For disaster evacuation:

- *If* arrive at the scene, *then* load the patients according to the evacuation triage rules.
  - Evacuation triage rules [111]: The on-scene victims are assessed and categorized into four levels by the triage standard described in subsection 3.1.1. An ALS ambulance can transport three patients at most, with one life-threatening and one severe patient in the back cabinet and one moderate patient on the front seat. Less severe patients can be transported in the place of more severe patients. For example, two severe patients can be loaded in the ambulance's back cabinet when no more life-threatening patients present. This situation is called evacuation

capacity transfer. A BLS ambulance can evacuate only one moderate-type patient at a time due to its limited transportation capacity and medical treatment capability.

- *If* loaded with patients, *then* travel to a selected hospital or medical facility. For a large-scale disaster event which involves mass casualties, it is recommended that patients be distributed to different hospitals rather than concentrating on one or only a few facilities because the hospitals can become saturated easily [110].
- *If* arrive at the hospital, *then* unload the patients.
- *If* finish unloading patients at the hospital, *then* process and travel back to the scene.

For normal emergency response: In the simulator, the normal emergency events are randomly generated according to certain call distributions. The corresponding responses are provided on a first-come-first-serve basis, by the closest and available (idle) resource at the time. Assume only one EMS ambulance is dispatched to one normal emergency call. Some of the EMS responses to normal emergency events are described in subsection 3.1.1.

- *If* arrive at the scene and the patient is assessed as less severe levels, *then* treat the patient at the scene without further transportation.
- *If* arrive at the scene and the patient is assessed as severe or above levels, *then* stabilize and load the patient.
- *If* loaded with patients, *then* travel to the closest, available hospital or medical facility.
- *If* arrive at the hospital, *then* unload the patient.
- *If* finish unloading patients at the hospital, *then* process and travel back to the base.

It is worth noting that during a major disaster event, the reserve EMS resources may be dynamically relocated to better utilize them to respond to the normal emergencies. A separate

EMS location optimization model is developed and integrated into the disaster simulation system [29]. The ambulances' near-optimal bases and service areas are calculated by the program. The location model is a separate research topic and will not be described in detail in this dissertation.

Fire Personnel and Equipment:

Fire personnel are the first responders in many areas. They can be dispatched immediately to respond in most situations.

- *If* dispatched, *then* process (assemble driver, firefighters and equipment) immediately.
- *If* dispatched and ready, *then* travel to the scene.
- *If* arrive at the scene, *then* stay and deal with the situation until further instructed. While performing tasks at the scene, the fire units can impact the scene in the following ways:
    - Positive impacts: Besides handling the fire situations, the firefighters can normally assist in some medical first-aid work such as simple trauma treatment and cardiac stabilization because they are trained for certain basic emergency medical procedures. Thus, a major positive impact of fire personnel presenting at the scene is that the deterioration rate of patients will be lessened.
    - Negative impacts: Due to the large vehicle size, fire trucks can affect the traffic flow significantly and/or interfere with other agents. For instance, if a street is blocked by the fire trucks, the EMS ambulances are not able to turn around at the scene quickly after loading patients so the victim evacuation will be retarded.
- *If* finish the current task and no further instructions, *then* travel back to the base.
- *If* dispatched to another mission, even in the middle of the current duty, *then* start a new cycle.

Hazmat Personnel and Equipment:

The hazmat operational rules are similar to the fire rules except that they are specially dispatched only when contaminated or hazardous substances are involved.

The above responders' operational rules are executed after the responders are dispatched. The dispatching decisions are made by the commanders and they can be optimized dynamically by the methods described in the later chapters.

4.3 FEATURES

Several important implementations of the D^4S^2 simulator are discussed specifically in this section, including the scene victim degradation model, scene congestion model and traffic model.

4.3.1 Scene victim degradation model

One of the most important factors we need to consider in the disaster response study is the victim degradation at the scene, especially for the seriously injured, life-threatening (LT) type of casualties. The victims' health condition deteriorates if untreated; victims also interact with the responders who treat and evacuate them. We develop and implement the victim degradation model in two versions: closed form model and agent-based simulation model. The closed form model utilizes available macroscopic survival functions to model the victim deterioration; while the agent-based simulation models more subtle interactions between the victims and responders to simulate the victim degradation situations.

4.3.1.1 Closed form model

Failure rate or hazard rate is an important concept in reliability engineering theory as well as in other areas. It is the frequency with which a system or component fails, usually denoted in failures per time period [7]. The concept of failure analysis has been borrowed by medical scientists to model the survival probability of casualties, i.e., deaths in biological organisms over time.

Survival functions are a major aspect of interest for us to model the mortality of life-threatening casualties. Here, we only consider the degradation of life-threatening victims because they are much more serious than other types of patients and other degradation models (e.g., severe victims degrade to life-threatening) can be formulated similarly. We use the word "victim" or "casualty" to represent the life-threatening victim or casualty in this context. Usually denoted as *S*, the survival function is defined mathematically as:

$$S(t) = \Pr(T > t) \tag{4-1}$$

where t is the time parameter and T is the random variable for time of death. In words, the survival function is the probability that the time of death is later than some specified time t. The survival function is also called patient deterioration or decay function because the injured patient's survival probability usually decreases over time without or even with medical care. There are several distributions used in survival analysis among which the exponential distribution is the most common one used [16, 21, 23]. Although we use the exponential survival function throughout this research, the function can be changed to other functional forms for different scenarios where specific data become available. The exponential survival function is formulated as below:

$$S(t) = g^{-\lambda t} \tag{4-2}$$

where the survival probability is primarily a function of time t because it changes over time and g is a mathematical constant (called decay base) greater than one. The responders' treatment at the scene can improve the victims' survival, and λ is a parameter that captures such improvement. Hence, $\lambda = f(k)$ is a positive function and inverse proportional to the number of medical responders (denoted as k) who are dedicated to treating and stabilizing the patients at the scene. A reasonable λ function can be written in the form of:

$$\lambda = f(k) = a^{-k} \tag{4-3}$$

where a is a constant power parameter greater than 1. This function can describe the contribution of medical responders to patients' stabilization: with more responders at the scene to stabilize the victims, λ is smaller so that survival probability S is bigger. This trend satisfies the fact that medical responders' treatment activity can improve the chance of victims' survival. Since the responders may not all have the same treatment skill levels (e.g., ALS vs. BLS), the number of medical responders k can be weighted differently for different types of responders. For example, the function (4-3) can be changed to:

$$\lambda = f(\underline{k}) = a^{-\left(\sum_i w_i k_i\right)} \tag{4-4}$$

where w_i is the weight for type i responders and k_i is the total number of type i responders. More trained responders have larger weight w_i such that they have more positive impact on the victims' survival. The form of the λ function is subject to changes as more specific data are available, but it should always reflect the trend as illustrated above.

In equation (4-2), let $b = g^{\lambda}$, we have:

$$S(t) = b^{-t} \tag{4-5}$$

where b denotes the decay characteristic of the survival probability adjusted by the treating medical responders' efficacy; b is greater than one in nature.

Based on the exponential setting, we want to show a property of the survival function and use that neat property to incorporate the time-continuous survival decay function into the discrete event simulation framework. Figure 4-6 illustrates an exponential survival decay curve. In the beginning (time = 0), there are v casualties on scene. The whole time period is divided into a series of small time intervals δ_t. The number of deaths in an arbitrary time interval is depicted by the red solid line in Figure 4-6 and it can be calculated by

$$d = vb^{-t} - vb^{-(t+\delta_t)} \tag{4-6}$$

The proportion of deaths during this interval to the surviving people in the beginning of this interval can be expressed as

$$p = \frac{d}{vb^{-t}} \tag{4-7}$$

$$p = 1 - b^{-\delta_t} \tag{4-8}$$

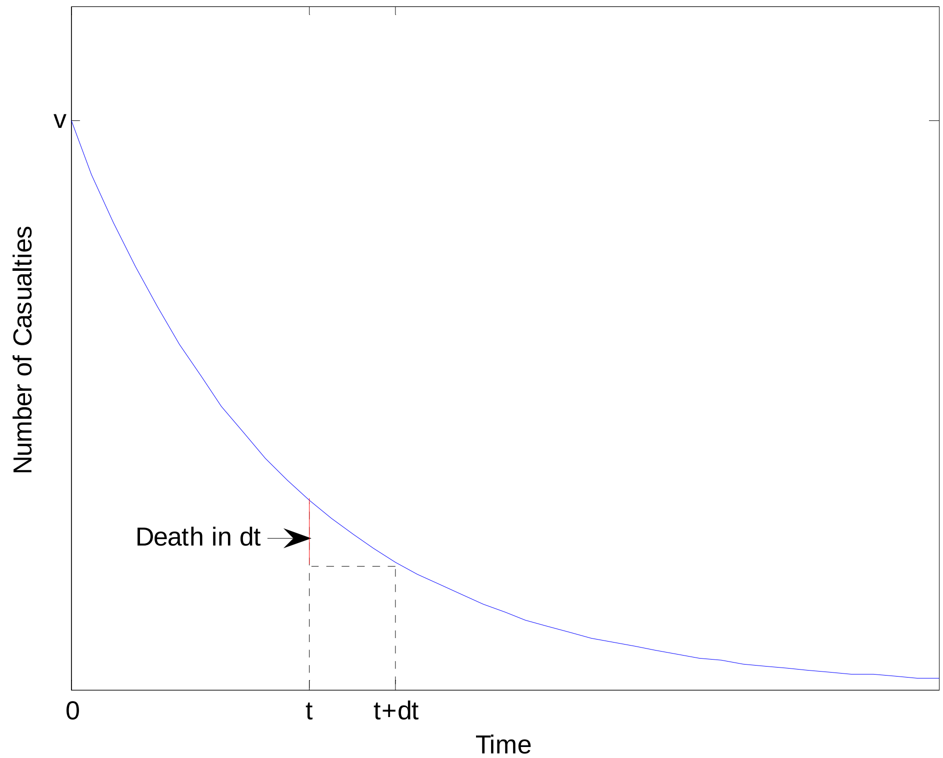

Figure 4-6. Illustrative exponential survival curve

It can be observed from equation (4-8) that the death proportion is only related to the decay characteristic parameter b and the length of the divided time interval δ_t. The parameter b is given as an input of the simulation system. As described in subsection 4.2.2, in order to synthesize the agent-based modeling with discrete event simulation, we need to discretize the time-continuous models by dividing them into small-time-interval events. The derived equation (4-8) provides us a theoretical base to implement the discretization idea.

During execution, the simulator keeps track of the number of surviving victims by subtracting the deceased and evacuated patients from the total surviving victims. It divides the continuous time horizon into small updating steps δ_t (e.g., δ_t = 5 minutes). At each step, it calculates the number of deaths in the next time interval using the equation (4-8). The total number of deaths is accumulated step by step. This closed form model assumes that no victims are evacuated during a time interval δ_t when the number of deaths is updated. The evacuation occurring during the current interval is recorded and reflected later in the next updating step. If the updating interval is small enough, the model can obtain substantially good quality results.

As the first approach to incorporating the scene victim degradation, we model all life-threatening victims as a group and only keep the count of the number of victims in the simulator instead of building every individual victim as an agent. This approach has the advantage of computational efficiency but it sacrifices the interactions between the victim and responder agents as will be addressed in subsection 4.3.1.2.

4.3.1.2 Agent-based simulation model

To implement the internal interactions between the continuously degrading victims and the responders, we extend the closed form model by formulating the casualties as individual agents and integrating them into the main simulator. As in the previous subsection, the life-threatening victims' deterioration is our main interest in this research.

The survival function defined in (4-2) is then changed slightly by eliminating λ because the stabilization efficacy of the disaster responders will be modeled in the simulator instead of being defined explicitly in the closed-form survival function.

$$S(t) = g^{-t} \tag{4-9}$$

Again, g can be any constant greater than one and it is not necessary to be the base of the natural logarithm; g captures the deterioration characteristic of the victims and it is an input parameter of the simulator.

The survival function is a time-continuous model. As stated in subsection 4.2.2, in order to incorporate a time-continuous model in the discrete event simulation framework, we need to discretize the continuous time into discrete, small time steps and model the events in a discrete fashion. Here, we consider this discretization approach first. Let $\lambda = 1$ and g be the base of natural logarithm e (standard exponential survival function), the survival function (4-9) is plotted in Figure 4-7. It shows that for a single victim agent, its survival probability (i.e., the probability that the patient is still alive at a specific time point) decays over time. That is, the figure demonstrates that at point $t = 1$, the victim's survival probability $S = 0.3679$, while at point $t = 2$, the probability that s/he is still alive declines to $S = 0.1353$ given that s/he did not decease until the point $t = 2$. In the agent-based simulation, we should check the status (i.e., alive or dead) of the individual patients based upon their survival probabilities at each time step t_i. For living individuals, they can be either stabilized at the scene or transported to a hospital; for deceased ones, their bodies will remain at the scene for future handling which is not included in the disaster response simulation. The victim status checking flow is depicted in Figure 4-8.

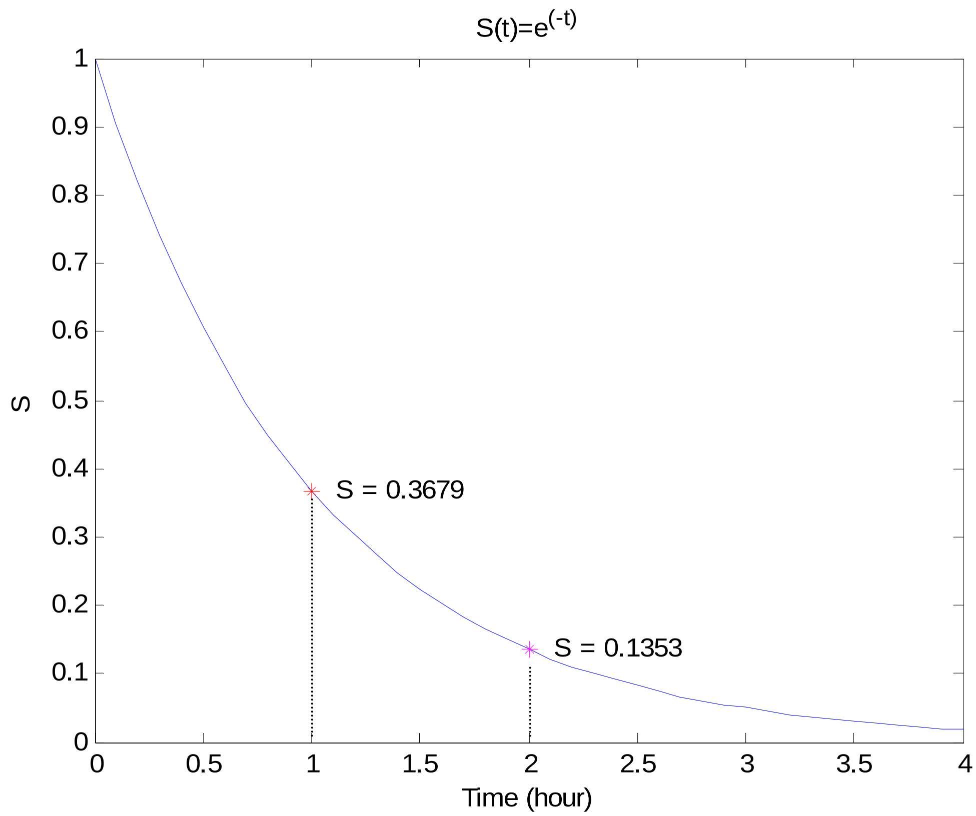

Figure 4-7. Plot of standard exponential survival function

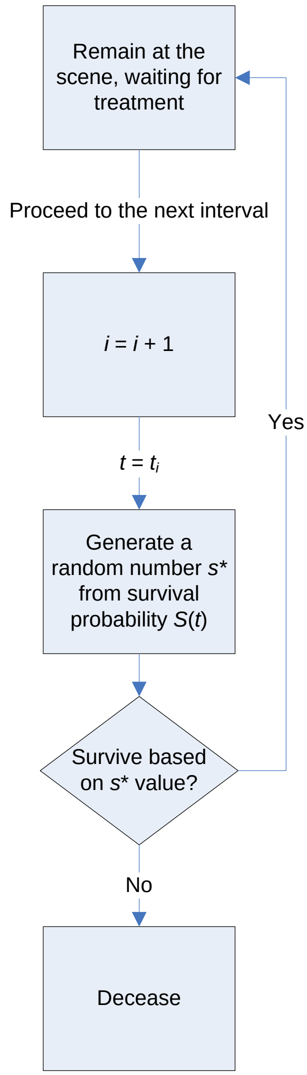

Figure 4-8. Victim agent status checking flow

While this appears to be a good approach, we soon discovered that the above status checking flow causes a serious problem that leads to the wrong survival probabilities for the life-threatening agents. A concrete example illustrates this flaw. For instance, a life-threatening patient Adam has the survival probability of 0.3679 at hour 1 according to the survival curve in Figure 4-7. We check his status based on the probability and find he is alive. At hour 2, the curve shows Adam's survival probability drop to 0.1353. We check his status again based on the new survival probability and find he is still alive. However, if the victim status checking flow is implemented in this way, 0.1353 is not the survival probability for Adam at the second hour as desired; instead it is the conditional probability of survival given the condition that Adam was alive in the first hour. The simulated survival probability at the second hour will be $0.3679 \times 0.1353 = 0.0498$ which is much smaller than the actual survival probability 0.1353.

It then becomes imperative to develop other approaches to model the degradation of the victim agents than simply discretizing the time-continuous survival function as commonly done to combine agent-based and discrete event simulations. An individual agent who is seriously injured can survive for a period of time until death. The length of survival time (i.e., time from injury to death) is an attribute of the agent, determined when the injury occurs and does not change over time, although such time follows a statistical distribution (e.g., exponential distribution) from the point view of the entire victim group. To avoid the probability problem described in the last paragraph, we convert the time-continuous survival probabilities to the survival time for each victim agent. Based on the survival time, we can then easily check the status of the agent in the simulator: if current time < survival time, the victim is alive; otherwise, s/he is dead. Because the survival time is an agent's attribute determined at the time of injury, it does not depend on the status checking scheme we use inside the simulator. In such a way, the real survival probability function is preserved statistically for the group of victims. This approach also has advantages in the discretization of the victim agent model. The status of the victims is checked only when one or more responders initiate interactions with the victims, e.g., treatment or evacuation.

To implement the approach, for each victim agent, we generate a random number s uniformly from the y-axis of the curve in Figure 4-7 and then map the number to the x-axis using the inverse survival function (for standard exponential survival function as shown in Figure 4-7):

$$t = -\ln s \tag{4-10}$$

The time obtained is the survival time for that specific agent based on the appropriate survival probability. We call this method the *Inverse Mapping Method* (IMM). It is developed to effectively convert time-continuous events (e.g., frequently checking the survival probabilities) to time to event (e.g., time to death). Time to event is also consistent with the principles of discrete event simulation. Therefore, with this method, we can easily incorporate the agent-based models into the discrete-event framework. A simple experiment is conducted to validate the approach. Suppose there are 160 victims at the scene and their health conditions deteriorate with the standard exponential survival function. Using the inverse mapping method, we can obtain the survival time (i.e., time to death) for each of the 160 agents and count the number of surviving victims over time. Figure 4-9 compares the IMM's and theoretical results which are shown to match well by eye-ball examination. In statistics, if fitting the standard exponential function to the IMM result data, the p-value of the F test turns out to be 3.4844E-114 which means the fit is significant. Further, the corresponding R^2 value is 0.962 that indicates a high goodness of fit level. We can conclude that they are statistically consistent with each other so the IMM method is valid.

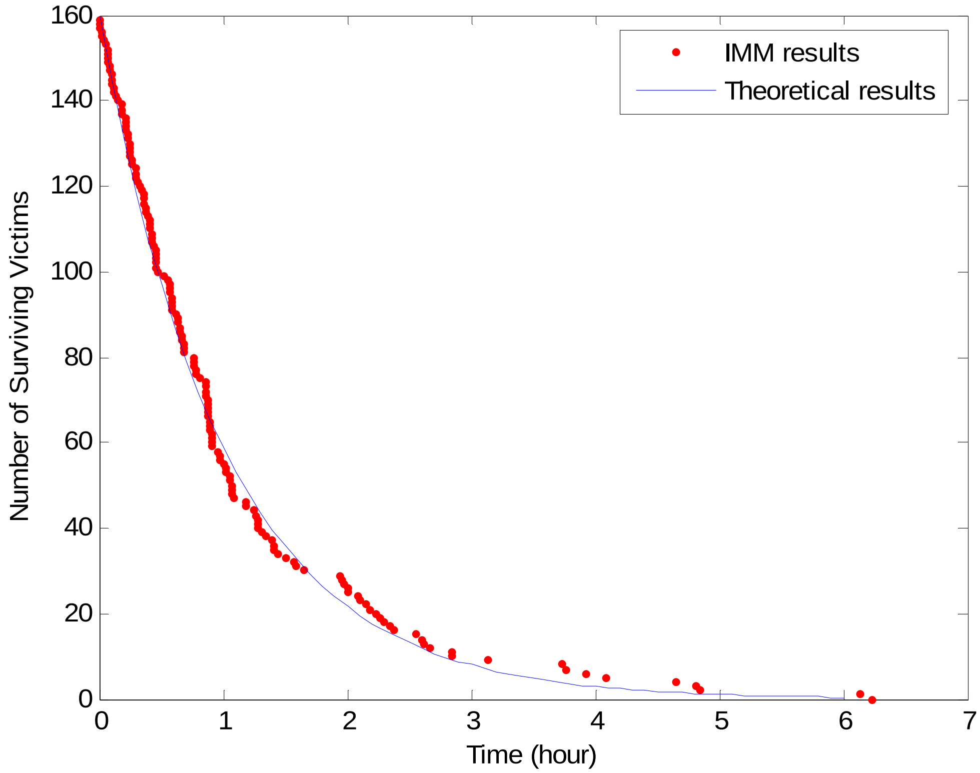

Figure 4-9. Experiment results for validating the Inverse Mapping Method (IMM)

Modeling the victims as individual agents and obtaining their survival times enable us to introduce more intrinsic interactions between the casualties and the disaster responders into the simulator. According to the agent-based framework, the victim status is expressed numerically and stored in a matrix as follows:

Table 4-3. Victim agent status matrix

Victim ID	Being Stabilized?	Time to Death	Dead?	Evacuated By Agent #
1	Y		N	
2	N	10 minutes	N	8

Table 4-3 (continued)

Victim ID	Being Stabilized?	Time to Death	Dead?	Evacuated By Agent #
3	N		Y	
4	N	40 minutes	N	
…				

Each row of the matrix defines the status of one life-threatening victim. Other types of agents (e.g., ambulances, medical responders) can access this table and choose the one or ones to interact with. The victim status values will be updated whenever they are changed by internal and/or external factors.

The rules and assumptions for life-threatening victim agents and the interactions with the disaster responders are defined in the following bullets. Although the rules are primarily designed for the trauma-type patients, they should also work for other types of victims and they are flexible to changes.

- A victim will survive at the scene without any medical care for a period of time defined by the original survival function. The victim will decease if no medical treatment or evacuation is provided before his/her survival ending time.
- All living victims wait for treatment in a queue in the order of ascending survival times. Every on-scene medical responder spends some time (e.g., 15 minutes, exponentially distributed) to treat and stabilize one victim at a time. The most serious but living patient (i.e., with the shortest survival time) obtains the highest priority for treatment, assuming

the triage personnel which is not part of the simulation can assess the severity of patients appropriately.

- Three different types of medical treatment are available: advanced-life support paramedics, basic-life support emergency medical technicians, and first responders (e.g., fire). Different responders have different skill and training levels for patient stabilization. As soon as the treatment is started on a living patient, the survival probability jumps to 1.0 (i.e., will not die) and then starts decaying after the treatment based upon an adjusted survival function, such that the patient is stabilized to some extent. Compared with the original survival function, the adjusted survival function usually has a smaller decay rate, i.e., smaller e value in equation (4-9), and the decay rate is also determined by the skill level of the responders. The most critical patient is always serviced by the best available emergency response person but once the treatment is started, the responder cannot be switched. After the treatment is done, if the victim is not evacuated from the scene (i.e., evacuation resource is not available), s/he then remains at the scene and waits for further treatment or evacuation. Because of the continuous medical stabilization, the survival curve is sawtooth in shape as shown in Figure 4-10 where medical services are available at time points a and c, and the treatment is done at time points b and d, respectively. The medical services provided by different personnel may vary and different survival decay rates are obtained as shown in the figure, blue curve segment from b to c vs. red curve segment after d.

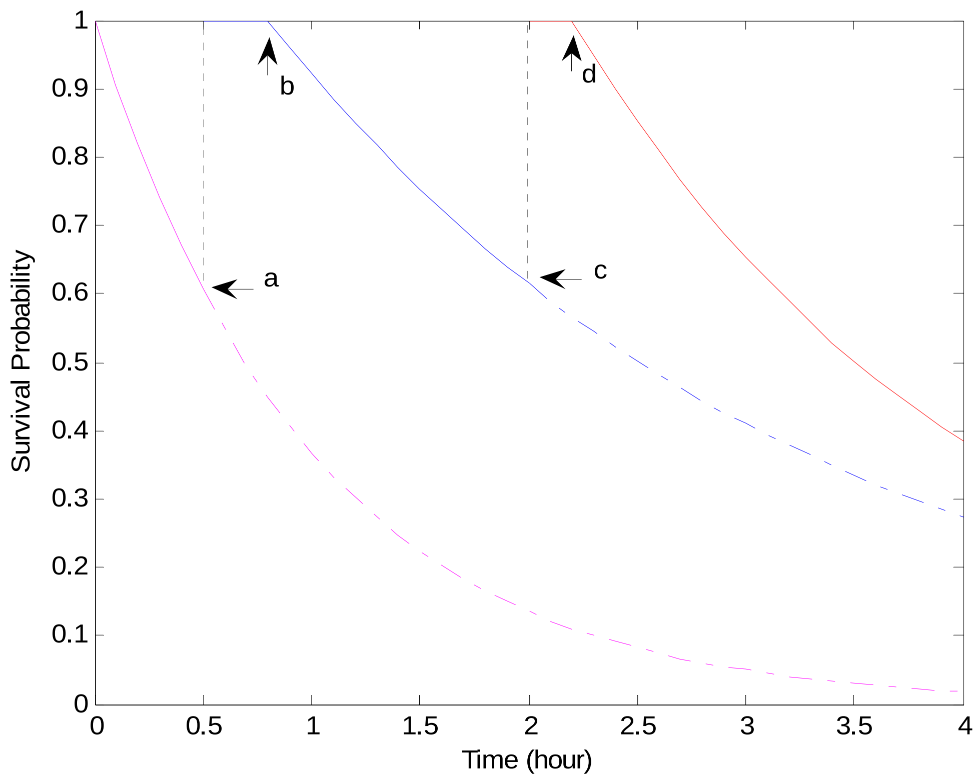

Figure 4-10. Sawtooth-shape survival curve for an on-scene patient

- Evacuation responders (e.g., advanced-life support ambulances) can transport the victims to hospitals or other medical facilities for further treatment. Each vehicle can take one life-threatening patient at most. Before transportation, it has to spend some time at the scene to asses and stabilize the patient. The evacuators always pick the patient who is alive, most severely injured and not being serviced by others at the time. The evacuated patients will not expire enroute until they arrive at a hospital for further service (post-hospital care) which will be accounted in the future extended simulation model.
- Evacuation has higher priority than on-scene medical treatment. When evacuation and stabilization services are both available for a patient, the patient is evacuated.

4.3.2 Scene congestion model

In a major disaster, when many emergency vehicles and resources present at the disaster scene in addition to the normal traffic, the scene could become congested quickly, especially by such oversize equipment as fire engines. The congestion can delay the responders' activities at the scene including treatment and evacuation of victims so it should be considered in the disaster simulator. The scene congestion situation is indeed affected by many factors such as vehicle density, weather, and infrastructure conditions.

Unfortunately, disaster scene congestion models do not exist in the current literature. One such model – Siren – is specialized software developed for emergency services by a group of researchers at the University of Auckland, New Zealand and Optima Corporation, which considers time-wise traffic congestion [74]. However, because the software has been commercialized, its details are protected and not available in the literature [75].

The disaster scene congestion mainly delays the emergency vehicles from accessing the scene. A well-structured, general traffic model could be a good substitute to model the scene accessing time when congestion occurs. We choose the modified Greenshields model for arterials [73] which relates the traffic speed with the vehicle density and road capacity. The model is presented as follows:

$$t_a = t_s \left(\frac{c}{c-n} \right)^{\alpha} \tag{4-11}$$

where t_a is the time for an emergency vehicle to access the disaster scene, t_s is the standard time (no congestion) to access the disaster scene, c is the scene space capacity at which the scene will be fully congested without any possible traffic flow, n is total space occupied by the responders at the scene, and α is power parameter. The literature [73] suggests a value $\alpha = 1.25$. The above scene accessing time model is implemented in both the disaster simulator and the optimization model presented later. It is worth mentioning that the function's parameters and even its form can be modified flexibly in other cases as needed.

4.3.3 Traffic model and vehicle flexible routing

The disaster response system is mainly modeled as a network-centric problem. The network structure in the simulation model represents the area's traffic system. Major intersections are chosen as the nodes and major roads and streets are chosen as the arcs. The nominal traversal time on each arc is calculated using the available traffic data such as speed limit and lane length. In the simulation, the responders travel along the network at the nominal speed under ideal traffic conditions. But the nominal travel speed may be lowered under the adverse traffic conditions due to many reasons such as inclement weather, construction and accidents. Extensive empirical studies on traffic flow in inclement weather have been published by the U.S. Department of Transportation Federal Highway Administration [54]. The speed reductions are presented in Table 4-4. The weather condition is incorporated into the disaster simulation as an input parameter to adjust the arc traversal times. This consideration enables us to simulate the responders' operations more realistically.

Table 4-4. Speed reductions in inclement weather [54]

Condition	Percent Speed Reduction
Dry	0%
Wet	0%
Wet and snowing	13%
Wet and slushy	22%

Table 4-4 (continued)

Condition	Percent Speed Reduction
Slushy in wheel paths	30%
Snowy and sticking	35%
Snowing and packed	42%

To route the emergency vehicles along the network, we use Dijkstra's algorithm [17] to find the traveling paths. Dijkstra's algorithm can quickly solve all the shortest paths from a given start point to all other nodes. When a responder agent decides to travel from one point A to another point B, and if the A-to-B route has not been computed previously, the algorithm is executed and all the resultant routes are saved in the database for future use in order to avoid unnecessary re-calculations. When the traffic conditions change (e.g., road closure) in the network, the affected arcs and routes are updated immediately and the traveling agents may be rerouted dynamically. This is called "vehicle flexible routing."

A large-scale, high-impact disaster event normally has the side effect of increasing the chance of other accidents in the region. The accidents at critical intersections or on major roads may also cause serious traffic congestion. Such congestion impacts the responders' travel times and delay the responses as a result so it has to be modeled in the disaster simulator. Table 4-5 records some congestion information needed by the simulator: node ID, start time and congestion type. Each row defines a congestion situation. Node IDs link to the congestion locations in the network. It is assumed that all the congestion occurs on the nodes (intersections) of the simulated network. If an accident happens on an arc (road) between two nodes, it is accounted to the nearest node. Different types of traffic congestion may correspond to different traffic models with different characteristics.

Table 4-5. Example table for traffic congestion information

Node ID	Start Time	Congestion Type
6	2:20pm	1
14	1:30pm	1
52	10:00am	2

When major traffic congestion occurs, the arc traversal times should be adjusted accordingly. In the simulator, at every update step, the adjustment factors are calculated and multiplied to the arc traversal times. We utilize an analytical macroscopic traffic congestion model internally as presented in the following. The travel times for the inbound traffic to the congestion are adjusted by multiplying a factor *f*. The outbound traffic is not affected significantly so it is not adjusted. The adjustment factors *f* can be modeled in the following way as an example.

$$f = \frac{r}{d} + 1 \tag{4-12}$$

$$\text{where, } r = r(t) = -a(t - t_p)^2 + p \tag{4-13}$$

If an arc is closer to the congestion's center, the arc traversal time is more impacted so the adjustment factor is larger. In equation (4-12), d is the distance from the arc's center to the traffic congestion's center and r is a parameter related to the elapsed time. After some time from the beginning of the accident, the traffic congestion reaches the peak and the traffic then goes back to normal gradually. Such a change in the traffic situation can be modeled by a second order polynomial function r with the boundary condition $r = 0$ when $t = 0$. In equation (4-13), t is the elapsed time from the beginning of the accident, t_p is the time when the congestion reaches its peak, p is the r's peak value, and a is a function parameter. Figure 4-11 and Figure 4-12 show an example of how the traffic adjustment factor changes with respect to the elapsed time and the distance to congestion center, respectively, according to the above model. At a specific location where the traffic is congested, the travel times keep increasing until a peak point at $t = 2$ and then decrease back to the normal state gradually. At a specific time when the traffic congestion presents, the travel times are increased more if the travel location is closer to the congestion center.

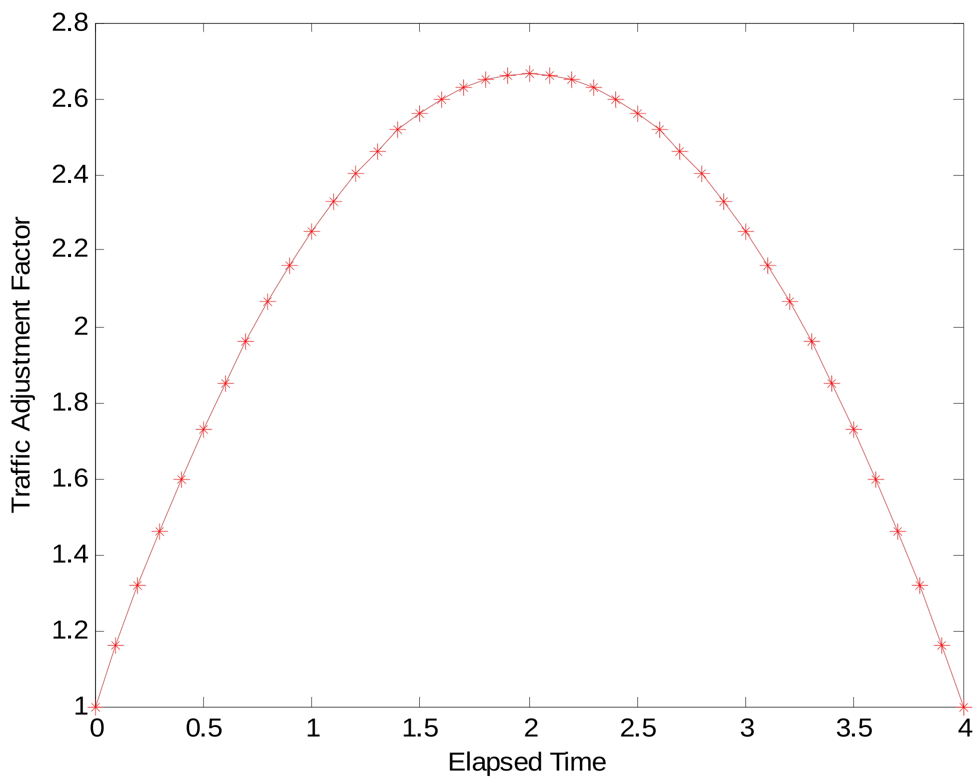

Figure 4-11. Changes of traffic adjustment factor with respect to elapsed time

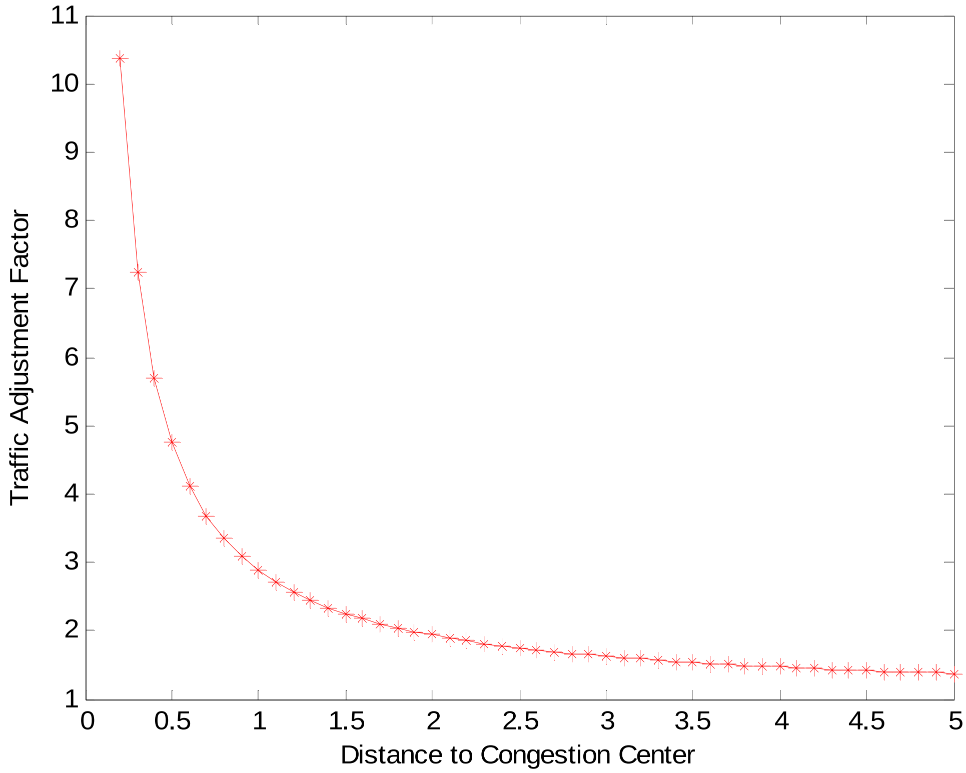

Figure 4-12. Changes of traffic adjustment factor with respect to congestion distance

Different types of traffic congestion have different characteristics and the parameters t_p, p and a should vary accordingly. The function $r(t)$ is time-continuous and it needs to be discretized as described previously. In the simulation, the traversal times on the arcs affected by the traffic congestion are changed dynamically and the corresponding agent routes are recalculated as well at each update step. Note that the traffic congestion model described in this subsection is a hypothetical but reasonable formulation. It can be replaced by other analytical or simulation models (e.g., [57] or [127], respectively) and integrated in our disaster simulation system when more specific traffic data are available. The sophisticated traffic modeling is a research area by itself and it is out of the scope of this dissertation.

4.4 COMPUTATIONAL RESULTS AND VALIDATION STRATEGIES

In this section, extensive computer experiments are conducted to test, verify and validate the system. Compared with the very first model presented in subsection 3.1.2, the current agent-based discrete simulation system incorporates much more realistic factors, flexibility and dynamics; in the meanwhile, the current, innovative model is more computationally efficient than before: one complete run with ten replications takes about 15 minutes compared to one hour previously. This performance shows the power of combining agent-based and discrete event simulation.

The large-scale, mass-casualty disasters are low-probability and high-impact events. Due to the lack of historical data or comparisons to the actual systems, the disaster simulation is extremely hard to fully validate. Several validation methods are explored in this research, in order to validate the system from various angles. This validation process is one of the major contributions of this dissertation.

4.4.1 Environment

The following computational tests were performed on a personal desktop computer with the Intel Pentium 4 CPU at 3.06GHz and 1.00GB of RAM memory. The operating system used was Windows XP. The simulation model was implemented in Rockwell Arena 10.0 with default settings [97].

4.4.2 Component validation

The simulation model itself consists of components that include victim degradation, responder actions, scene congestion and vehicle routing submodels. Interactions exist among the different components but as a first effort, we can isolate them and validate some of the critical individual components individually.

The agent-based victim degradation model presented in subsection 4.3.1.2 is an excellent example component to validate in this research. In the model, the life-threatening patients at the scene decay to death overtime based on a specific survival curve. The on-scene medical responders treat and stabilize them in the rule-based fashion so the condition of the patients can be improved with the responders' involvement. To validate the process, we need to run several experiments to investigate the model's behavior.

To simplify the problem and make the results more intuitive, we assume the on-scene stabilization medical responders are available immediately at the beginning of the simulation runs and they are all at the same medical treatment skill level, i.e., after treatment, the patients have the same decay base *e* defined in equation (4-9). To run the victim degradation submodel as a component, we implement it separately from the complete D^4S^2 model and run several experiments for different numbers of on-scene medical responders. In Figure 4-13, the number of deaths is plotted against the elapsed time for different scenarios with zero to six available medical responders treating the patients at the scene.

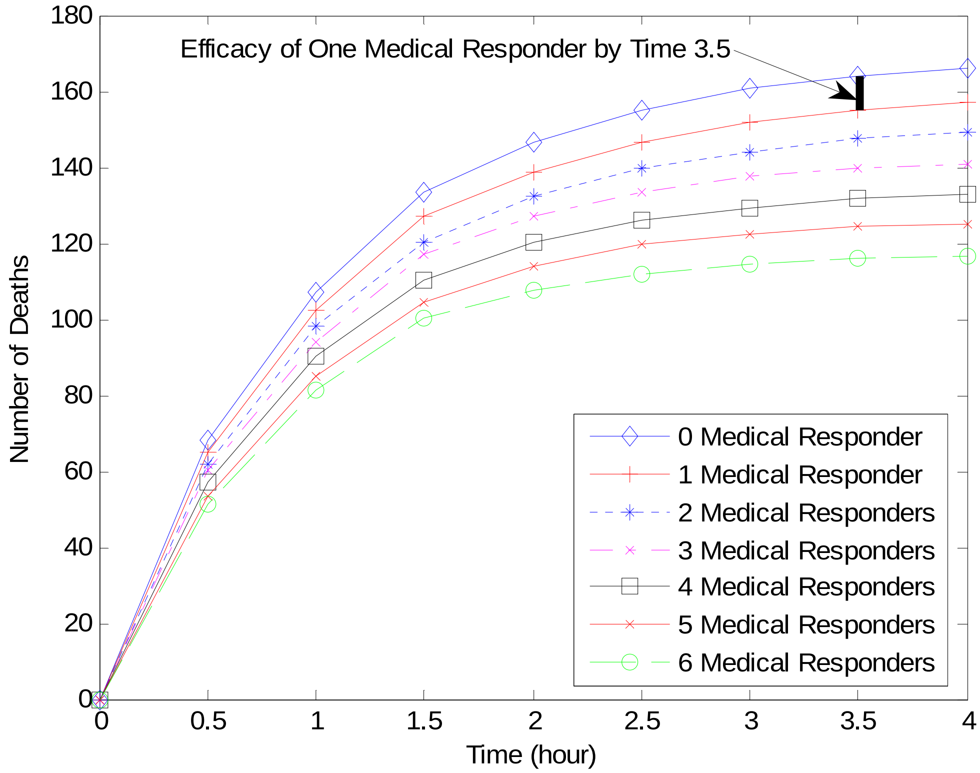

Figure 4-13. Impact of on-scene medical responders to victim degradation

As noted in the above figure, we can define the medical responder's efficacy as the total number of lives saved by the on-site responder by a given point in time. Such efficacy increases monotonously over time. At the first glance, we may find that the impact to victim degradation (i.e., medical responders' efficacy) is linearly related to the number of responders. The rough intuition behind it is that the life-saving capability increases when more resources are available. This observation is further validated by analyzing the simulation results shown in Table 4-6.

Table 4-6. Simulation results of number of deaths

Medical Responder# / Time	0	1	2	3	4	5	6
0.5	68.15	64.8	61.95	60.2	57.1	53.65	51.3
1	107.2	102.55	98.1	94.05	90.15	84.8	81.2
1.5	133.3	127	120.45	116.85	110.35	104.45	100.05
2	146.55	138.9	132.15	127.2	120.05	114.2	107.55
2.5	155.1	146.7	139.65	133.55	126.1	119.5	111.6
3	160.95	151.9	144.05	137.45	129.3	122.25	114.45
3.5	164.2	154.95	147.45	139.55	132.1	124.25	116.2
4	166.2	157.2	149.15	140.8	133.15	125	116.65

Suppose the efficacy of one medical responder (i.e., unit efficacy) is known by subtracting the 3rd column from the 2nd column in the above table and denote the unit efficacy as x_1 . We can obtain an approximately linear relationship between the number of medical responders and their efficacy from the above results as:

$$x_n = anx_1 \tag{4-14}$$

where x_n is the efficacy of n medical responders by a point in time, a is a constant coefficient and n is the number of responders. Based on the above simulation results, we get the constant parameter a = 0.910 with a standard deviation of 0.0588. Similar results can be obtained for other settings of the experiments.

In order to validate the model, we want to show the intuition behind it. We assumed that all the on-scene medical responders work at the same skill level so that they spend the same amount of time to treat every assigned patient. To better illustrate the problem in numbers, we assume it takes each medical responder 15 minutes to treat and save one life-threatening patient. If only one responder works at the scene, one patient will be saved after 15 minutes, two patients will be saved after 30 minutes, and so forth. If two responders are available at the scene, in the first treating period (i.e., first 15 minutes), two patients can be saved; by the end of the next period, in total four victims are saved, and so forth. It is obvious that the efficacy is linearly related to the number of responders with parameter a = 1.0 (a is defined in equation (4-14)) in this ideal case. However, simulation is a stochastic system instead of a deterministic, ideal one and it also involves many other factors. Considering the noise and other minor interactions in the simulator, the results of the above experiments satisfy the intuition behind them so that they can be used to validate the agent-based victim degradation submodel to a reasonable extent.

4.4.3 System validation

Once the important components are validated, the complete simulation system can be run as a whole to test the extent that it can represent the actual system. This process is called system validation. Since conducting the actual, physical experiments on such high-impact events as disasters is impossible and the historical events are rare or not well documented, we have to use different approaches to validate the system indirectly.

4.4.3.1 Experimental validation

Experimental validation has its roots in experimental science. It is the most popular and accurate method to validate a simulation system. It basically runs the simulation and actual physical system under the same input and environment and compares their performance measures against each other. The simulation is valid if it behaves identically, within certain error ranges, as the actual system does; otherwise, its validity should be challenged.

The classical experimental validation is hard to implement for high-impact event (e.g., mass-casualty disaster) simulations because the destructive event cannot be created in reality due to the ethical considerations. However, experiments can be conducted for the counterpart low-impact events or for only a portion of the entire system. Take the disaster response problem as an example. We can "make" a disaster event but without real injured casualties and let the responders react as if the event really happens. This experiment, at least, can validate the traffic patterns during a major disaster event. Small-scale experiments can hardly help validate the simulation under extreme conditions. Sensitivity analysis needs to be performed to investigate the abnormal (e.g., nonlinear) behavior of the system when its scale exceeds certain levels.

4.4.3.2 Theory based validation

The idea of theory based validation is to utilize some theoretical or analytical scenarios and results to test the validity of the simulation model. To do this, some well-structured scenarios to which the theoretical framework can be applied have to be designed and simulated. Sensitivity analysis is normally used for this validation method. It can be conducted by changing one independent variable while fixing all the other independent parameters and observe the change in dependent (i.e., response) variables.

Real cities have complex, non-symmetric infrastructure and resource distributions which make the disaster responses behave nonlinearly. Such nonlinearity is hardly predictable by theoretical methods. This is the reason why we develop the simulation system to help predict behaviors. However, a symmetric square flat city could be a toy model for us to apply simple, existing theories to validate the simulation to some extent. The square flat city is a special case of a more complex city.

An artificial square flat city network having 15 by 15 grid nodes is designed as in Figure 4-14. The nodes are arranged as a perfect grid matrix and they are connected by vertical or horizontal arcs. The following experiments involve the sensitivity analyses in which we vary one factor (variable) while fixing the rest and observe how the responses change accordingly.

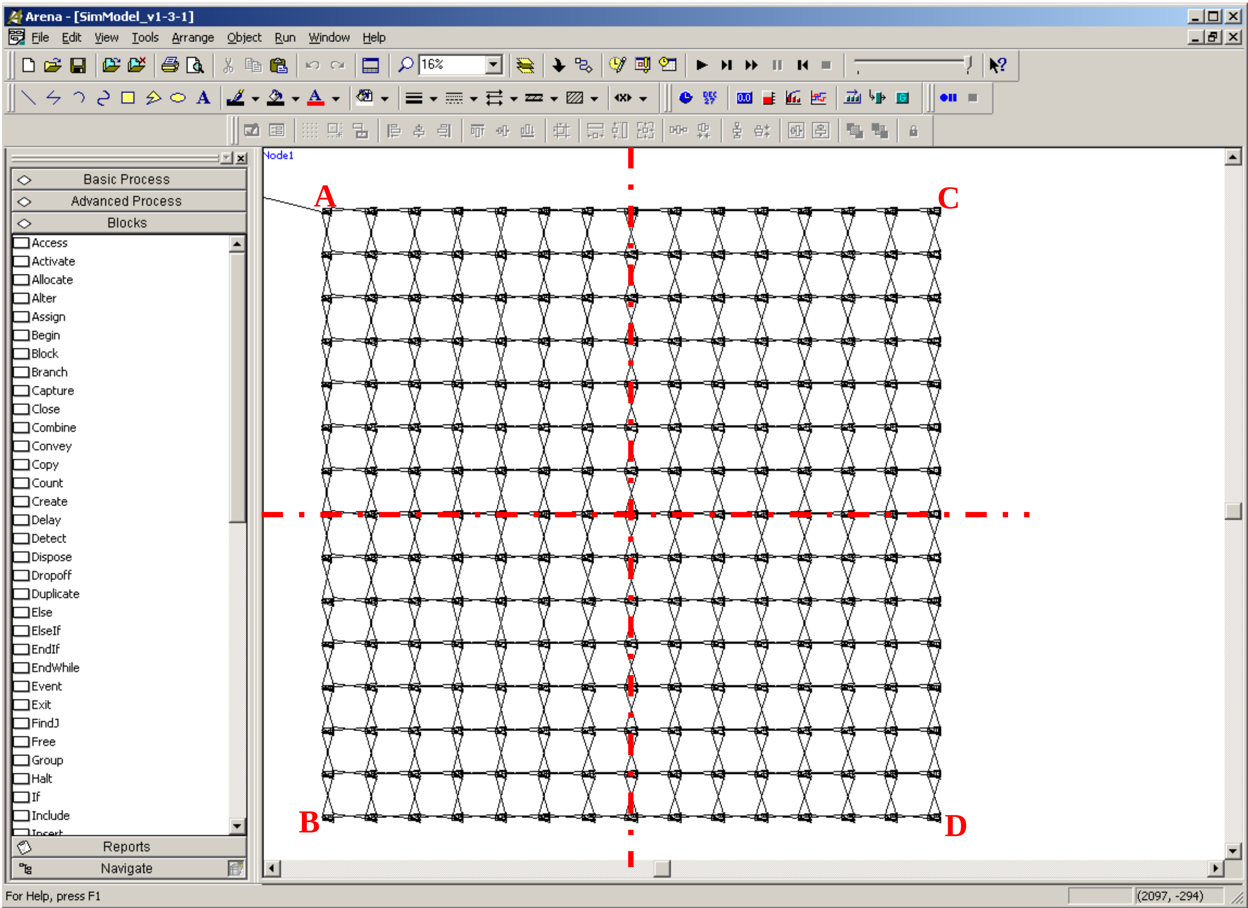

Figure 4-14. 15 by 15 square flat city model in Arena

Experiment 1:

Suppose the disaster happens at the center node of the network. The event has 100 life-threatening, 80 severe, and 80 moderate victims who need to be treated and evacuated. Eight ALS responder vehicles are dispatched to evacuate the victims and eight first responder vehicles (i.e., fire trucks) are dispatched to treat and stabilize the patients at the scene. Four hospitals A, B, C and D are located at the four corners of the city so that they have the same distance to the disaster scene. For evacuation, we assume the hospitals have unlimited capacity and the victims are distributed to the four hospitals equally, i.e., in the order of A-B-C-D-A- ... Since the topography and topology are perfectly symmetric, the complete 15 by 15 network can be reduced to less than one quarter as in Figure 4-15. The reduced network comprises the scene node, several nodes around the scene and the hospital node at the corner. Those nodes between hospital and scene area are omitted and replaced by just an arc (bold in the figure). The reduced network can be used to simulate the full network much more efficiently. Hence, the upper right quarter of the original network is simulated.

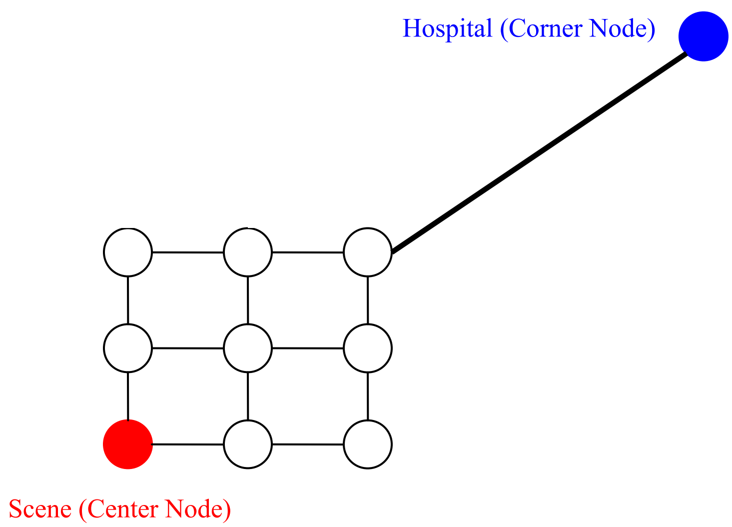

Figure 4-15. Reduced square flat city network

In this experiment, we vary the hospital distance to the disaster scene (i.e., the length of the bold-black arc in the above figure) to investigate the response results. Figure 4-16 shows the relationship of average ALS responders' round-trip time vs. hospital distance. The hospital distance is measured by the travel time between the hospital and the scene. The round-trip here comprises three parts: traveling from scene to hospital, unloading patients at the hospital, and traveling from hospital back to the scene, excluding the scene operations such as loading patients. It can be observed from the figure that the round-trip time increases linearly with the hospital distance and the intercept equals to the hospital unloading time (23.16 minutes/ALS vehicle). This result is validated by the following equation.

$$t_r = t_u + 2 \cdot t_h \tag{4-15}$$

where t_r is the EMS round-trip time, t_u is the constant unloading time at the hospital and t_h is one-way travel time from hospital to the scene.

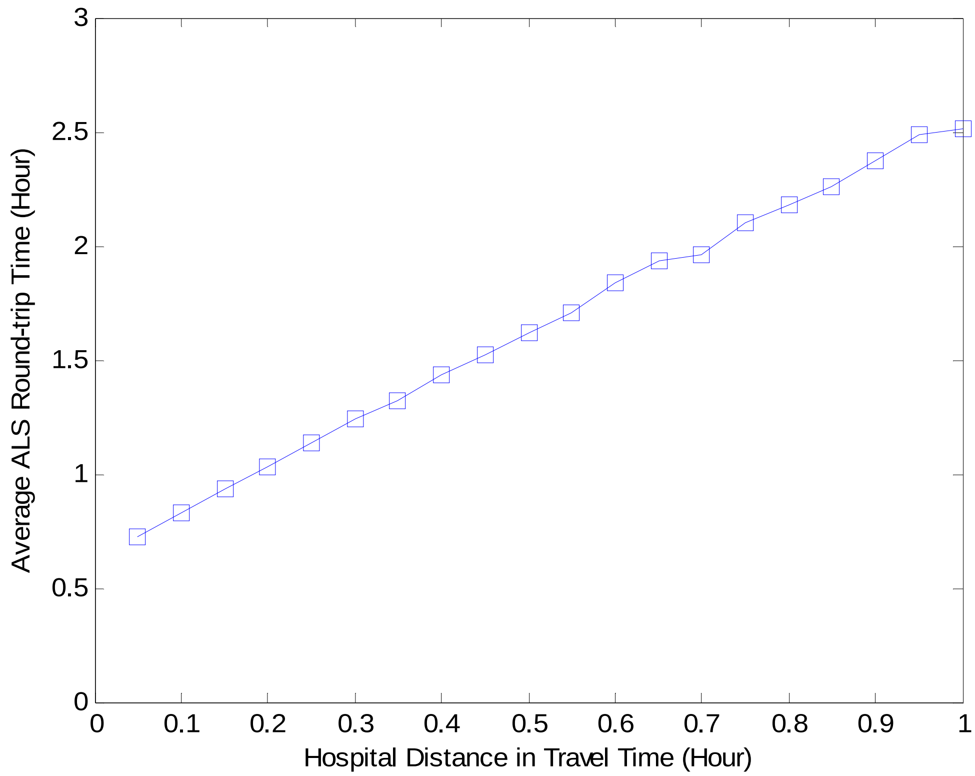

Figure 4-16. Square flat city experiment 1: Average round-trip time

Figure 4-17 shows the relationship of scene clearance time of the three types of patients vs. hospital distance. The severe and moderate patients' deterioration is not considered in this experiment so their clearance time increases linearly with the hospital distance due to the linear relationship between ambulances' round-trip travel time and hospital distance as shown above.

Some life-threatening victims decay to death according to the survival distribution. Such deterioration becomes greater when the hospitals are located farther away from the scene because the transportation times are longer and the necessary treatment is delayed. This implies that more fatalities result and, unfortunately, fewer life-threatening patients need to be evacuated. For example, according to the computational results shown in Figure 4-17, if the hospital travel time increases from 0.3 hour to 0.8 hour (i.e., from 18 minutes to 42 minutes), the increase in moderate and severe patients' evacuation times are 7.44 hours and 7.06 hours, respectively, but the increase in life-threatening patients' evacuation time is 4.52 hours which is a significantly smaller. Thus, the life-threatening victims' clearance time increases more slowly with the increase in hospital distance compared to severe and moderate patients' situations.

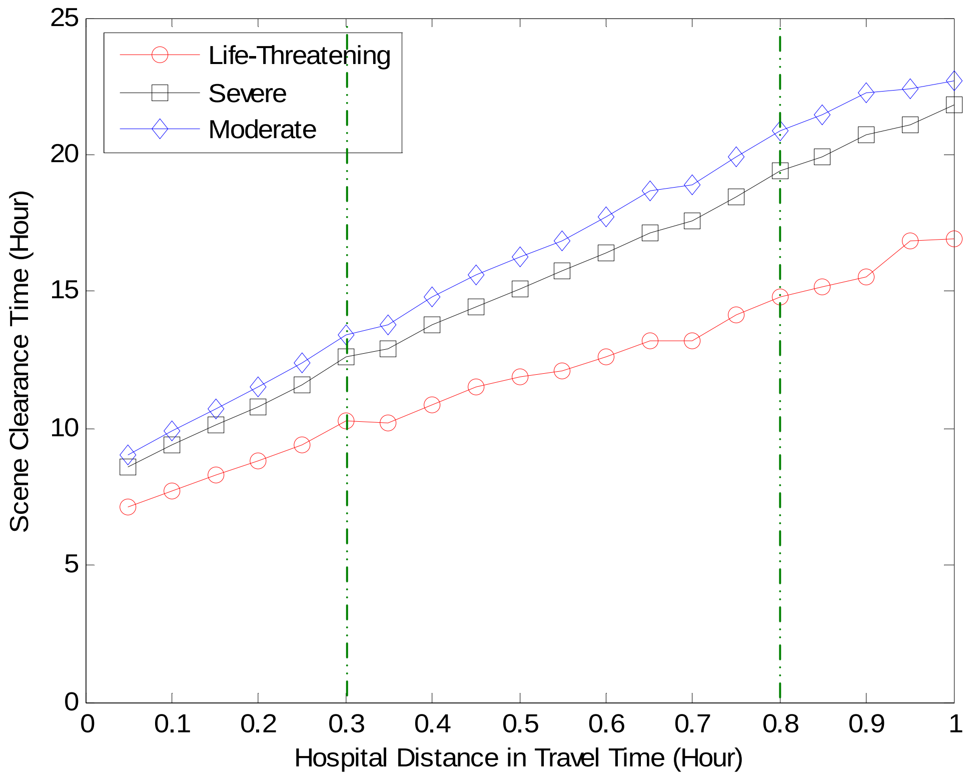

Figure 4-17. Square flat city experiment 1: Scene clearance time

Figure 4-18 shows the relationship between the number of deaths and hospital distance. The blue solid-square curve is the actual simulation results and the red smooth curve is the polynomial fit line. Note that the simulation results fluctuate a bit because of the random noises in the stochastic simulation. The number of deaths increases with hospital distance but not in a linear fashion. When the evacuation takes more time, the chance of saving more life-threatening victims will decrease because they cannot survive longer at the scene without treatment. The curve is also consistent with the fact that the life-threatening decay is less sensitive to the hospital distance when the hospital is located farther away from the scene.

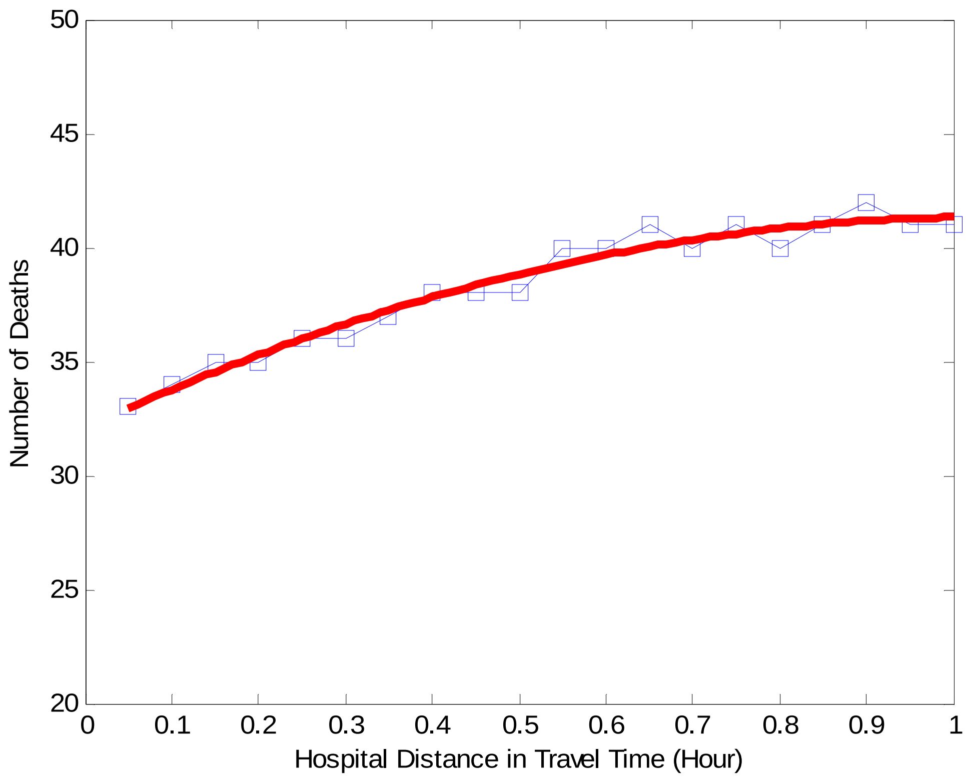

Figure 4-18. Square flat city experiment 1: Degradation of life-threatening patients

Experiment 2:

It would be more interesting to scale up and down the size of the disaster to compare responses. Following the above experiment, we run three more experiments: the first one has 50 life-threatening, 40 severe, and 40 moderate victims; the second one doubles the victim quantity, having 100 life-threatening, 80 severe, and 80 moderate victims; the third one doubles the victim quantity again, having 200 life-threatening, 160 severe, and 160 moderate victims.

Figure 4-19 and Figure 4-20 show the comparisons of the severe and moderate total evacuation times, respectively, among the three experiments. All of them linearly increase with hospital distance and are proportional to the corresponding number of patients.

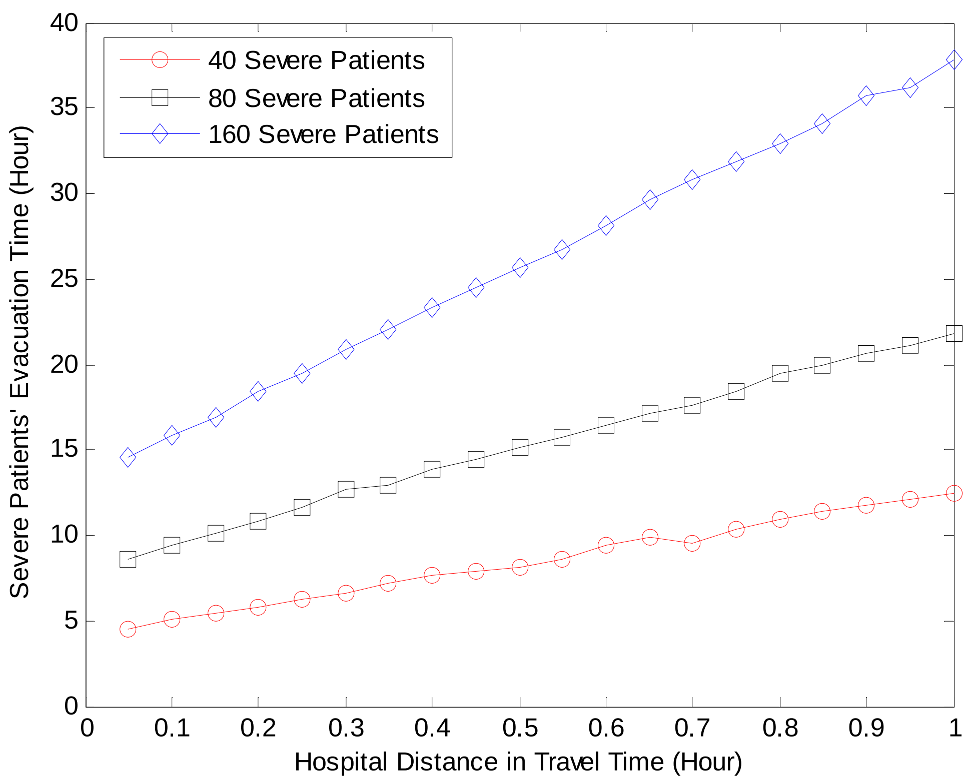

Figure 4-19. Square flat city experiment 2: Severe patients' clearance time

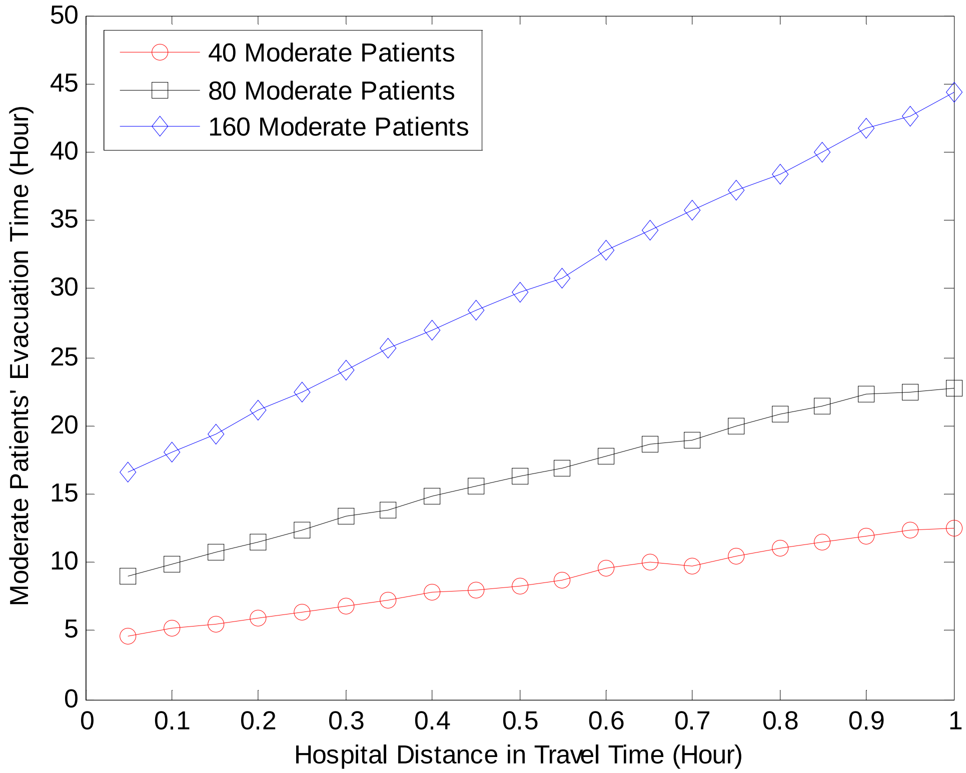

Figure 4-20. Square flat city experiment 2: Moderate patients' clearance time

Figure 4-21 and Figure 4-22 present the life-threatening (LT) evacuation time and death number at the three scales: 50 LTs, 100 LTs, and 200 LTs, respectively. The LT evacuation time is not proportional to the number of LT patients as it is for the above severe and moderate cases; the increase in evacuation time is slower than the increase in the number of patients. It is not surprising that with more victims at the scene, the responder resources become much more overwhelmed and deaths increase more rapidly as illustrated in Figure 4-22. The more deaths that are created, the fewer patients need treatment and evacuation.

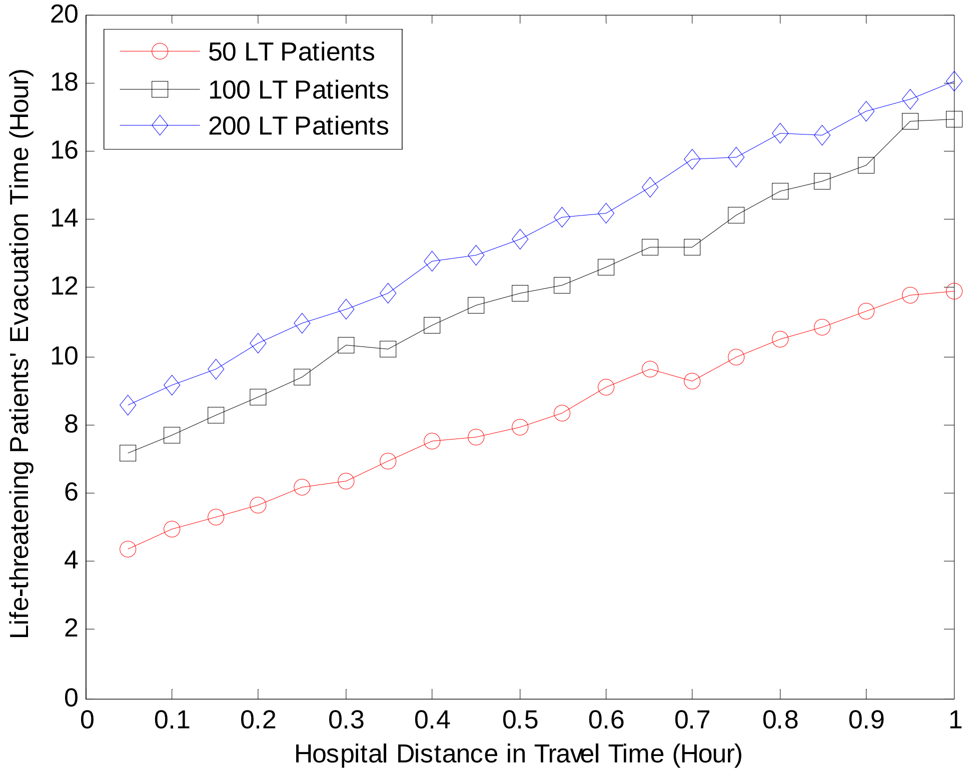

Figure 4-21. Square flat city experiment 2: life-threatening patients' clearance time

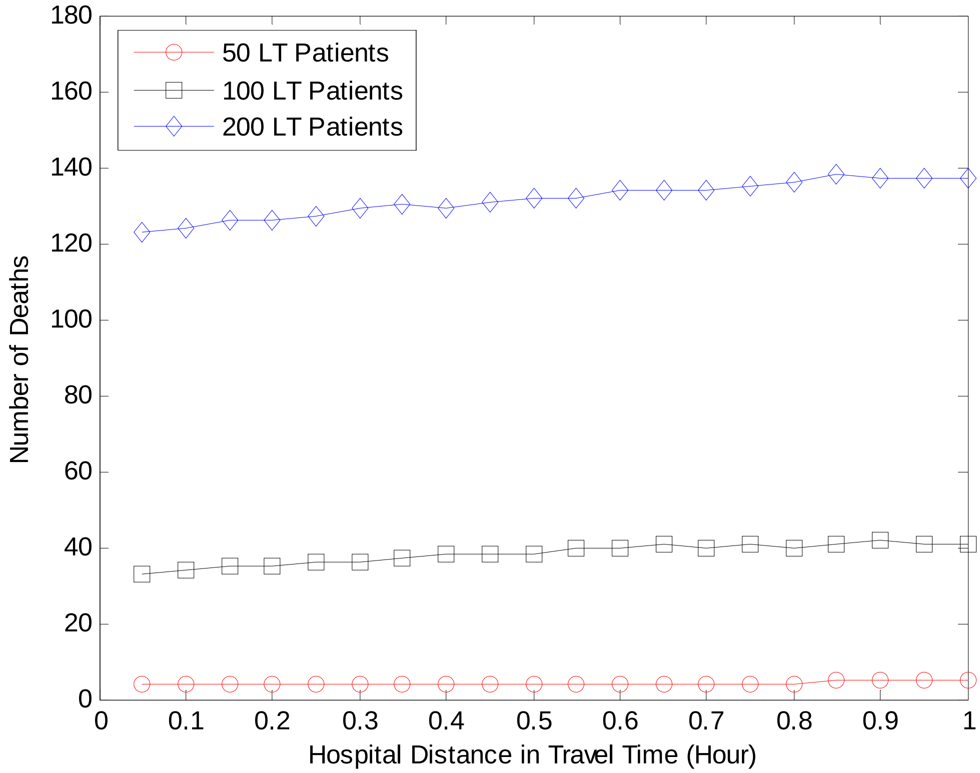

Figure 4-22. Square flat city experiment 2: Degradation of life-threatening patients

Experiment 3:

This experiment breaks the symmetry of the network and investigates the changes in responses. The reduced network depicted in Figure 4-15 is not applicable, so we must use the original full model to conduct this experiment. Suppose the event involves 200 life-threatening, 160 severe, and 160 moderate victims. The four hospitals now have unbalanced capacity. Refer to Figure 4-14. The left two hospitals A and B have limited capacity, each having 20 beds for life-threatening, 30 beds for severe, and unlimited moderate capacity; while the right two hospitals C and D have unlimited capacity. The responders still attempt to distribute the patients to the hospitals in the order of A-B-C-D-A-… as long as the capacity allows. In this experiment, we vary the scene location along the horizontal center line (the horizontal red dashed center line in Figure 4-14). The scene location is expressed by its relative distance (i.e., traversal time) to the network's center, negative to the left and positive to the right.

Figure 4-23 (blue solid-diamond line) shows that the ALS ambulance's average round-trip time decreases linearly when the scene location is farther away from the left-hand side where limited-capacity resources locate. The finding is consistent with the theoretical foundation described below.

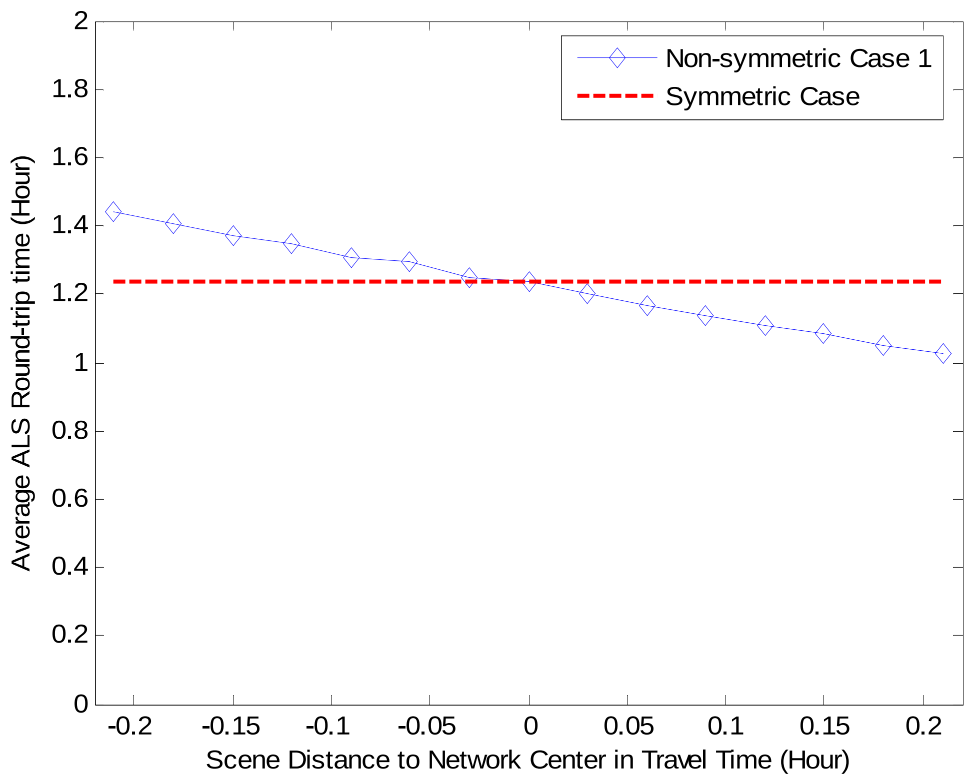

Figure 4-23. Square flat city experiment 3: Average round-trip time

Suppose n_1 patients are evacuated to the left-hand hospitals and n_2 patients are evacuated to the right-hand hospitals. The distance from hospital A (on the left) and hospital C (on the right) to the scene location is denoted as x and y, respectively. Since the scene location is sliding from the left to the right on the horizontal center line, the total distance $(x + y)$ equals L

which is a constant. This is a fundamental phenomenon in optics. The average ambulance's round-trip time can be expressed and manipulated as below (t_u is the unloading time at the hospital).

$$
\begin{aligned}
t_r &= \frac{n_1(2x+t_u)+n_2(2y+t_u)}{n_1+n_2} \\
&= \frac{n_1(2x+t_u)+n_2(2(L-x)+t_u)}{n_1+n_2} \\
&= \frac{2(n_1-n_2)}{n_1+n_2}x+t_u+\frac{2n_2L}{n_1+n_2}
\end{aligned}
\qquad (4\text{-}16)
$$

In this experiment, the number of deaths decayed from life-threatening patients is about the same across all the runs so n_1 and n_2 are constants and $n_2 > n_1$. The resultant function in (4-16) represents a straight line with a negative slope which complies with the simulation results. In Figure 4-23, a horizontal line (red dashed line) passes through the central point's result. It represents the situation of the symmetric network model as presented previously in Experiment 1 and 2. Figure 4-24 shows the scene clearance time vs. scene location. Similar to the above linear relationship, the scene clearance time is also linearly related to the distance from the disaster site to the network center point.

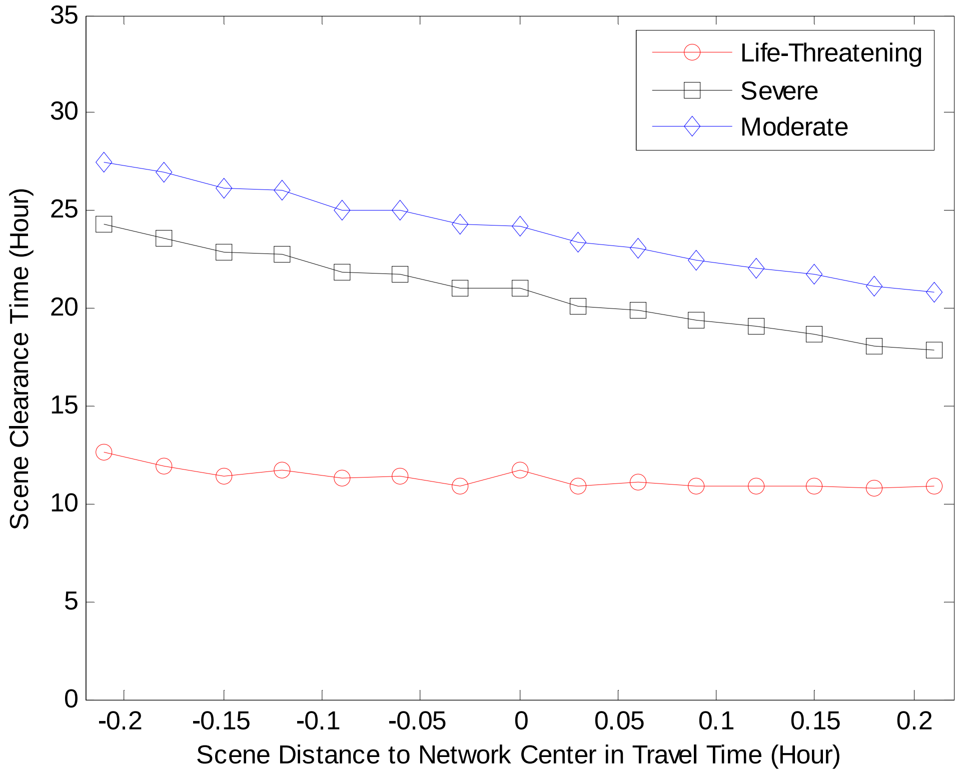

Figure 4-24. Square flat city experiment 3: Scene clearance time

Experiment 4:

In all of the previous experiments, the number of responders is fixed. This experiment varies the number of EMS ambulances to investigate the scene congestion's impact on the response. When a large number of emergency vehicles converge at the disaster scene, the traffic congestion will increase. The modeling of the scene congestion is presented in subsection 4.3.2. The responders' time to access the scene in this experiment is plotted in Figure 4-25. It shows that when the number of ALS vehicles exceeds 45 or so, the scene is seriously congested and the time to access the scene increases much more rapidly, even up to several hours. In this extreme case, the responders can hardly enter the scene to treat and transfer the victims in time.

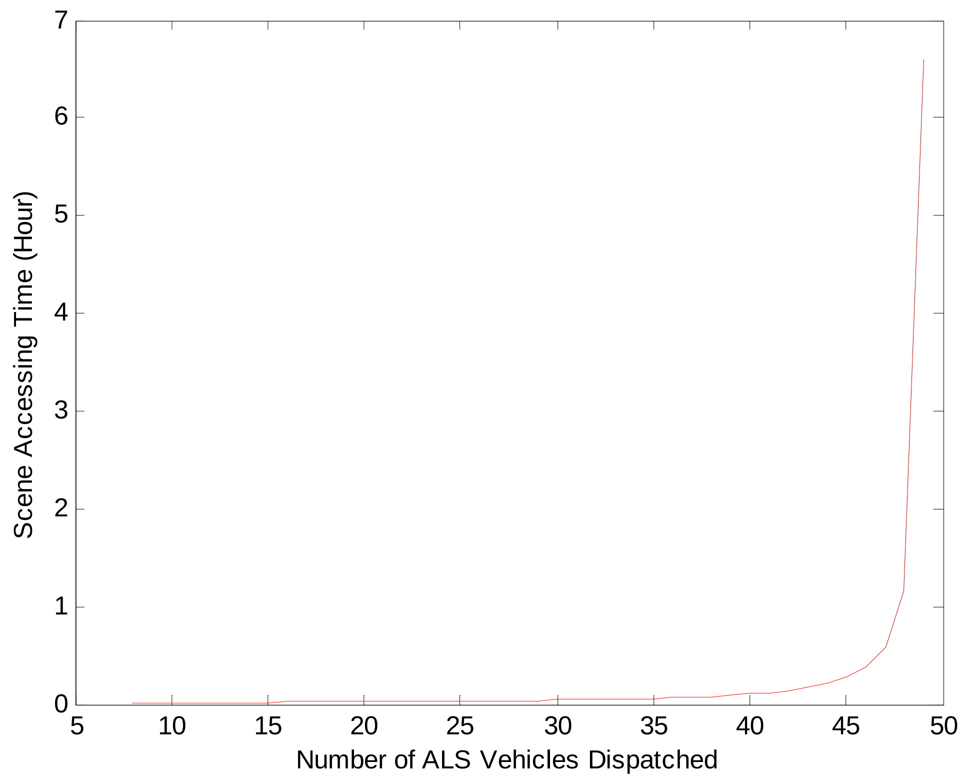

Figure 4-25. Square flat city experiment 4: Scene accessing time

Figure 4-26 compares the evacuation times for life-threatening, severe, and moderate victims under conditions (1) without scene congestion considered and (2) with scene congestion considered, respectively. Before the scene becomes overwhelmed, i.e., the dispatched ALS vehicles are fewer than 45 or so, the victim evacuation times for the two situations, i.e., without and with congestion, are similar. If the scene becomes saturated, the congestion effects appear obviously: the victim evacuation times increase dramatically because the responders cannot access the scene easily and the responses are delayed. All of the three plots in Figure 4-26 show the tail-up curves of scene clearance time when the number of ALS vehicles is larger than 45, which are consistent with the scene accessing time model presented in Figure 4-25.

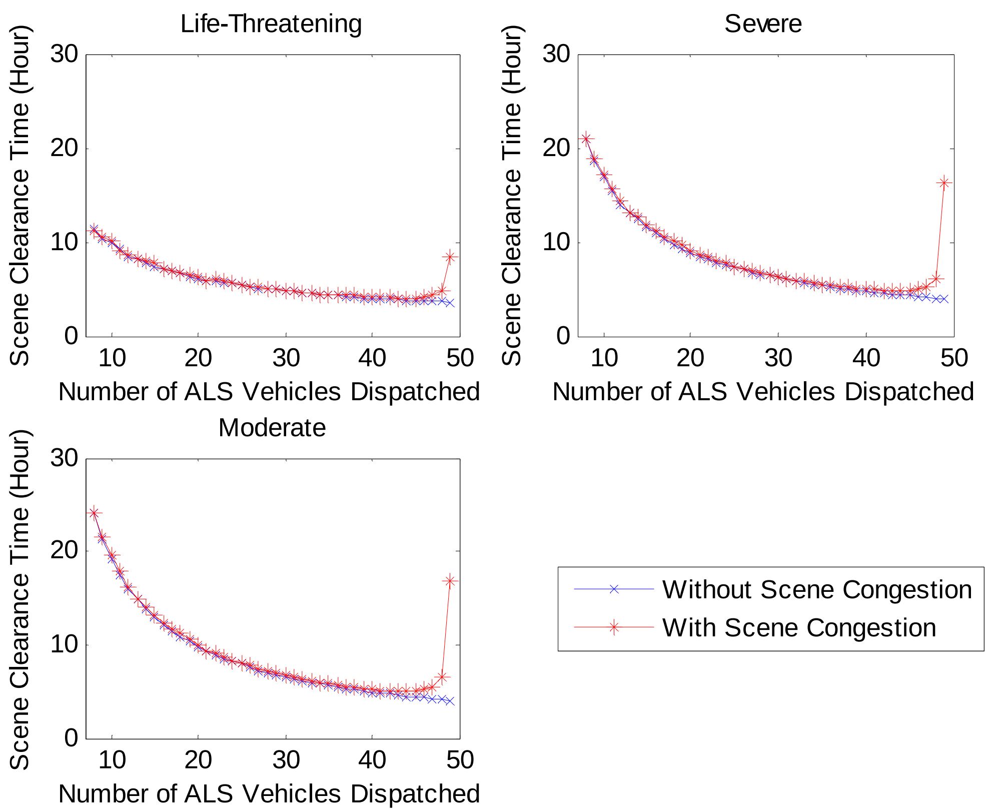

Figure 4-26. Square flat city experiment 4: Scene clearance time

When the scene is highly congested, the responders' travel and tasks will be delayed so much that more victims at the scene will decay to death before being evacuated. Figure 4-27 compares the no-congestion and congestion situations in terms of the fatalities among life-threatening patients. The end of the star-marked red curve shows a dramatic increase in the number of deaths where the scene is highly congested with too many emergency vehicles and the treatment and evacuation of the victims are delayed. The same phenomenon can be found in the previous results.

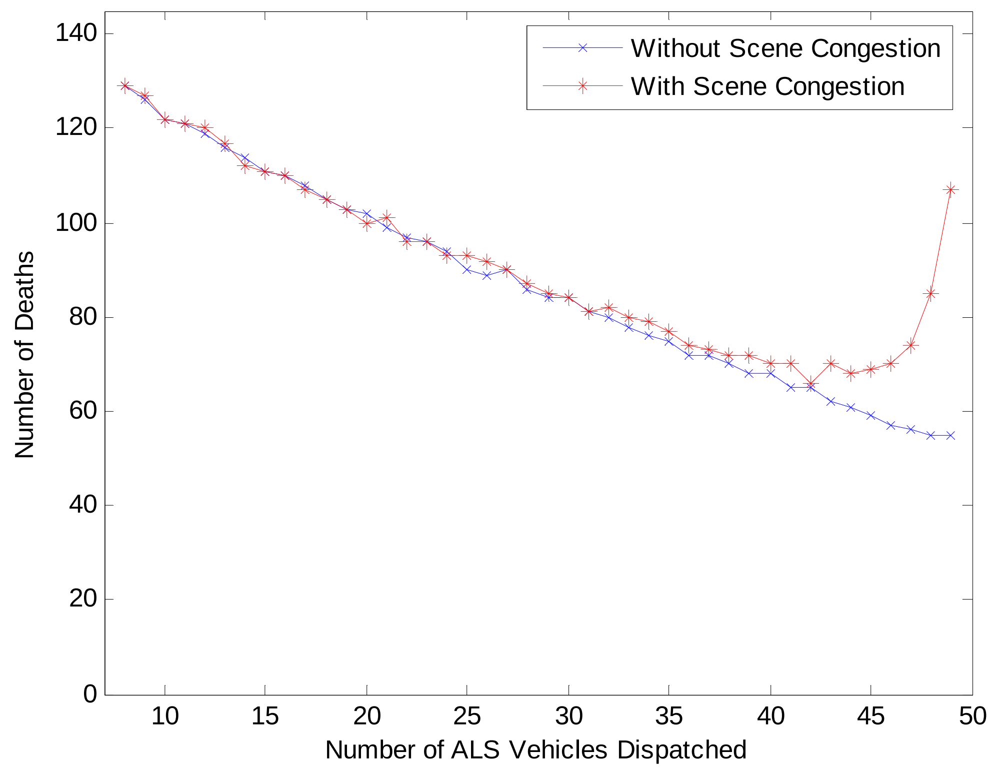

Figure 4-27. Square flat city experiment 4: Degradation of life-threatening patients

Experiment 5:

The previous four experiments all address the disaster response issues but do not consider the normal emergency responses. The loss of the coverage of normal events during a major disaster event may also lead to increased fatalities and morbidities, so it is not ethnically allowed. This experiment studies the performance of EMS responses to normal events (i.e., normal calls) with changing call volume. Assume the normal calls distribute uniformly in the square flat city network. Four hospitals are at the four corners with unlimited capacity. Eight EMS ambulances are involved in the responses. The performance for normal call responses is evaluated by the measurement called response degradation. Response degradation here is defined as the proportion (probability) of unsatisfactory responses, i.e., emergency calls in which the response time exceeds a certain target (e.g., x minutes). The x minutes here is essentially the targeted service level of the response system. For a specific targeted service level, the smaller the degradation value, the better response performance achieved. For a fixed degradation value, the higher the targeted service level (i.e., smaller x value defined above), the better response performance achieved. In this experiment, we assess the response degradation on five targeted service levels: x = 8, 12, 16, 20, and 24 minutes, respectively.

Figure 4-28 shows the relationship of normal call degradation vs. call volume for the different targeted service levels. The response performs better in terms of the degradation values if the targeted service level is decreases. For example, the response degradation values are smaller at targeted service level of 24 minutes compared to those at targeted service level of 8 minutes because more responses are assessed as “satisfactory” within 24 minutes. When the targeted service level is fixed, the degradation value increases but more and more slowly with the increase in call volume. The curves are asymptotic to 0 when the call volume is low because all of the emergencies can be handled successfully; on the other hand, the curves are asymptotic to 1 when the call volume is high because if all of the emergency responses are unsatisfactory, the response degradation value equals to 100% but cannot exceed 100% according to its definition. All of the observations described above are valid mathematical trends, consistent with intuition and the way we model the system, so they can help further to validate the simulation system.

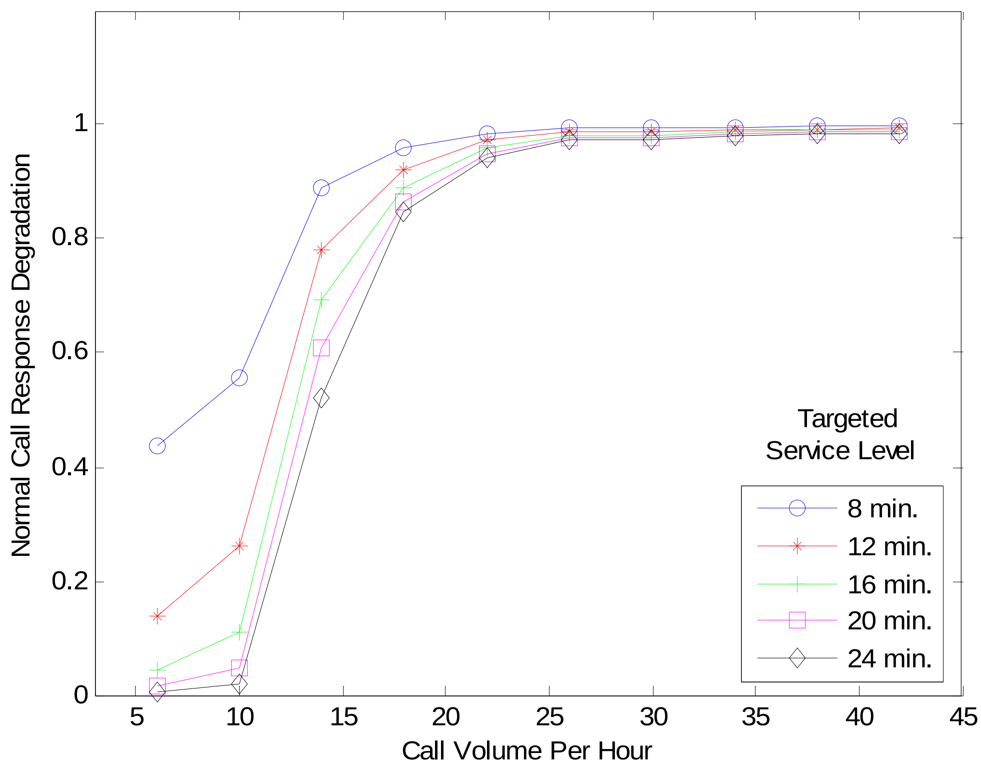

Figure 4-28. Square flat city experiment 5: EMS normal call response degradation

4.4.3.3 Historical validation

Historical validation uses archival data to replicate past scenarios in order to test the validity of the simulation. Historical events can provide real situations and operations which give additional confidence to the users as well as achieving face validity. However, the past events are often not recorded with enough details (i.e., data are incomplete) or they might not reflect the current configurations of the system. Furthermore, the data collection and documentation for low-probability and high-impact events is normally a retrospective process. In other words, the archived data were collected after the events so they may not truly represent the situations during the events. Thus, using such historical data may flaw the studies.

4.4.3.4 Exercise validation

Exercise validation is a robust method to test the synthesized system as the whole. It executes pre-written, "realistic" role playing scripts and the simulation model in parallel in order to compare their outcomes as an assessment of validation. To some extent, the exercise is analogous to computer simulation. However, some researchers think the exercise scripted scenarios are created artificially and may not resemble reality [106]. Furthermore, exercise participants usually do not behave exactly as they should do in actual disasters because they know the exercise is just an artificial practice. All of these factors may compromise the realism of exercises.

4.4.3.5 Subject matter experts validation

The subject matter experts (SMEs)'s professional experience is of great value for system validation. Several experiments are designed and presented to SME resources for their expert opinions which can be used for improvement or confidence in the simulation model.

Two of such experiments are based in Pittsburgh, PA. The experiments focus on validating the critical infrastructure utilized by the disaster responders in the area. The utilizations of street intersections are calculated by the counters set up inside the simulation. When an emergency vehicle passes a particular intersection during the simulation run, the corresponding counter is incremented by one. The utilization results are then reflected on the city maps using the spatial analysis tools embedded in the GIS system. This capability demonstrates the beauty of integrating simulation with GIS by enhancing visualization and analysis of the results.

Figure 4-29 shows the experiment result of a major disaster event happening on the north side of the city (marked by a yellow star in the figure). The critical locations for the response are color coded on the map. The red dots present the most important intersections which are utilized most frequently by the emergency vehicles to respond and transport victims. The orange dots are important and the yellow dots are less important intersections. They form a critical evacuation path going towards the eastern hospital locations which are marked by the blue-square hospital symbol on the map. Such a critical path provides very meaningful information to the decision makers by indicating the most important infrastructure for which they need to focus on improving the response efficiency.

Figure 4-30 shows the experiment result of another major disaster event happening in the west end of the city (marked by an orange star in the figure). The critical areas for the response are also color coded on the map. The red and orange areas illustrate the critical intersections and paths for victim evacuation. The paths start from the scene location, extend along the river and finally enter the downtown area towards the eastern hospital area.

Both experiment results were validated by Allegheny County Emergency Chief Robert Full [34]. Most of the critical intersections are identified and confirmed by him based on his twenty-years in the emergency response profession.

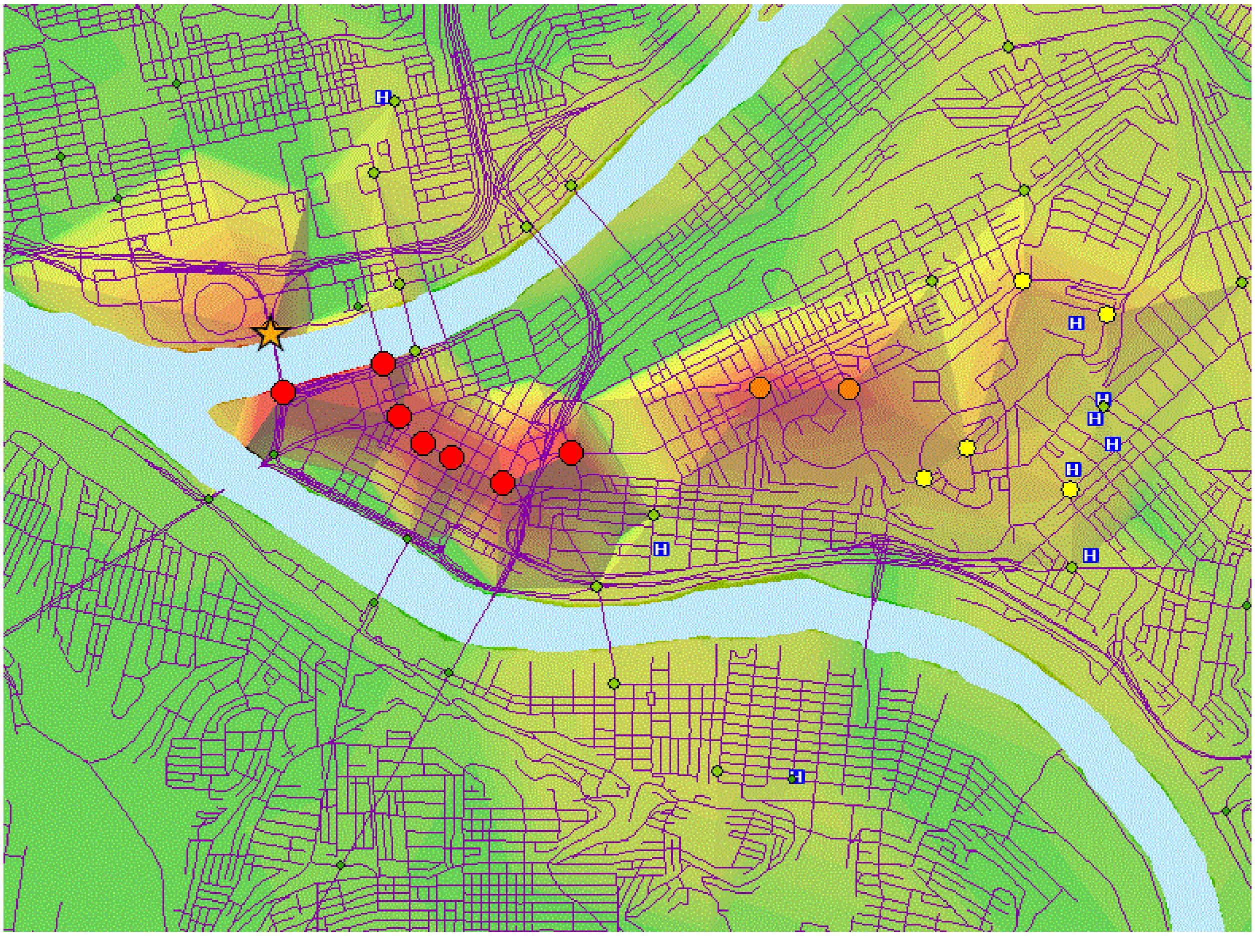

Figure 4-29. Experiment result of Pittsburgh north side disaster

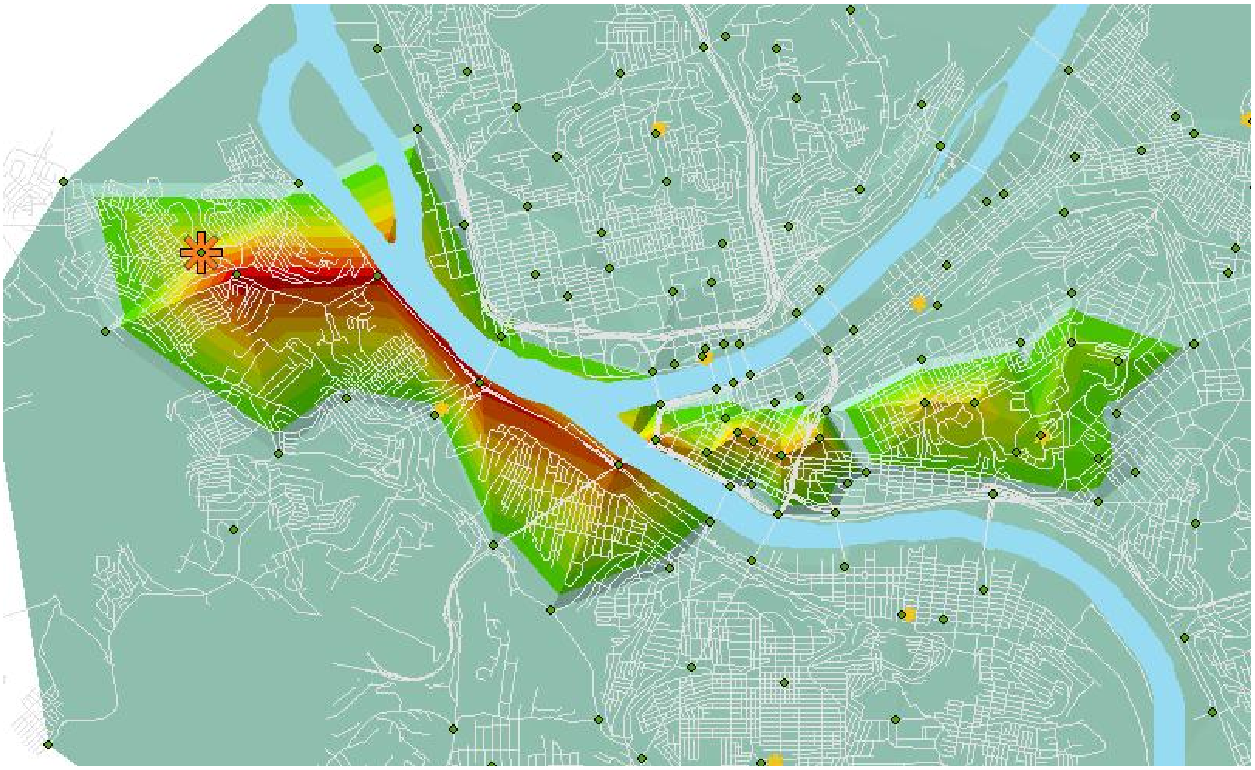

Figure 4-30. Experiment result of Pittsburgh west end disaster

To conclude the subsection of "system validation," the table below summarizes and compares the various system validation methods mentioned above.

Table 4-7. Comparison of simulation system validation methods [106]

	Experimental	**Theory Based**	**Historical**	**Exercise**	**Subject Matter Experts**
Mechanism	Operate the physical system and model in an identical scenario.	Use predicted behavior from theoretical frameworks (e.g. field theory).	Use data from previously recorded events.	Create a role playing script that parallels an event.	Present model runs to SMEs.

Table 4-7 (continued)

	Experimental	**Theory Based**	**Historical**	**Exercise**	**Subject Matter Experts**
Test	Compare metrics from physical system and model.	Compare metrics from theoretical framework and model.	Compare metrics from actual events and model.	Compare metrics from exercise and model.	SMEs evaluate the quality of the model.
Comment	This is the technique used to validate most simulations.			This is a well-accepted technique.	
Advantages	Most exactly tests the model.	Simple. Inexpensive.	Real.	Close to real. Robust.	Experience based. Predictive.
Disadvantages	Can't create a disaster just to validate the system.	Theoretical solutions are not robust enough to model actual situations.	Events are often not recorded with enough details. The portfolio of past events may not reflect what will happen in the future.	Expensive. Behavior in the scripted scenario may not reflect reality.	Expensive. Relies on the quality of the SMEs.

4.5 SUMMARY

The simulator is the central component of the Dynamic Discrete Disaster Decision Simulation System. In this chapter, the agent-based architecture is hybridized with a discrete event simulation framework in order to maintain the model's flexibility, dynamics and efficiency at the same time. The generic hybrid system modeling methodology is discussed in detail as well as the specific implementations for the disaster response simulation system. Both the effectiveness and efficiency of the hybrid disaster simulation system have been demonstrated through comprehensive computational results. The results also validate the correctness of the model from various perspectives.

The simulation system is only a descriptive tool without the capability of optimizing decisions by itself. The next chapter aims at incorporating heuristic-based optimization techniques into the simulator in order to extend it to be a decision tool.

5.0 EVOLUTIONARY REAL-TIME DECISION MAKING PROCEDURE

5.1 INTRODUCTION

Complex problems such as disaster planning and response management involve numerous stochastic factors which make any pure analytical method ineffective or inefficient. Simulation is an attractive approach for modeling large-scale systems due to its ability to realistically represent stochastic events.

Here, we want to borrow a psychology terminology – *Situation Awareness* (SA) – to describe our problem. The term SA was raised two decades ago and has been defined formally as the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [25, 26, 27]. A validated simulation model can be used to evaluate the current system and predict future situations by rationally projecting from the current to the future states. According to the definition, the simulation model itself can be looked at as a situation awareness tool. The question is then how to utilize this awareness tool to help make good decisions in order to better manage future outcomes.

Although simulation is powerful in modeling complex operational systems, it is basically a descriptive tool that is not designed for optimization directly. However, a number of researchers have developed various simulation-based optimization algorithms and procedures to use simulation to make better decisions (see examples in Chapter 2.0). Unfortunately, few have addressed the evolutionary nature of the decisions as well as the simulation itself. In the real world, the solutions to complex problems and systems are always changed to better adapt to new situations. Furthermore, the information observed or collected as the input parameters of the simulator is usually partial and incomplete in the beginning. As the event advances, more accurate information will be available and could be input into the simulation system sequentially. Thus, dynamically reevaluating and updating the decisions become necessary when the event unfolds gradually and better information and insights are obtained. Later decisions do not independently exist: they depend on earlier decisions and the changing situation. This is defined as an evolutionary decision process. In this research, a simulation-based heuristic approach to systematically generating time-dependent, good-quality solutions to complex systems is developed. The output of the procedure is an optimal stream of decisions instead of the traditional single solution. Such a procedure can be used in the real-time management of disaster scenarios.

5.2 EVOLUTIONARY DECISION GENERAL FRAMEWORK

Most large-scale systems and complex processes run for a long period of time – up to many hours, days, or even longer. The disaster response system is a typical example of such systems. A major disaster event normally involves a large number of victims that need to be evacuated; many parties and social sectors interact with each other so that the whole response process may last for hours to days. During the course of the event, no one decision is universally good for all scenarios at all times. It is not surprising that new, unexpected situations may arise as better information becomes available. Consequently, previous response decisions need to be re-evaluated and might have to be revised accordingly. For ad hoc, unexpected events such as disasters, the management cannot plan or experiment beforehand. In this case, managing the situations in real-time during the events becomes imperative. Figure 5-1 illustrates a basic simulation-based real-time evolutionary decision process. The essential concept is to break the entire process time horizon into a time-dependent series of stages (i.e., small consecutive time intervals) and make appropriate decisions for those small intervals separately. In every decision stage, new information and data are input into the system, the simulator is run based on the updated data, and simulation-based optimization methods are utilized to estimate the future situations and obtain a good solution (decision) for each possibility. The future action point at which we target should be a period away from the current time point because it takes some time for the system to run and we cannot alter the current or past actions as time goes on. The detailed procedure is described in the following.

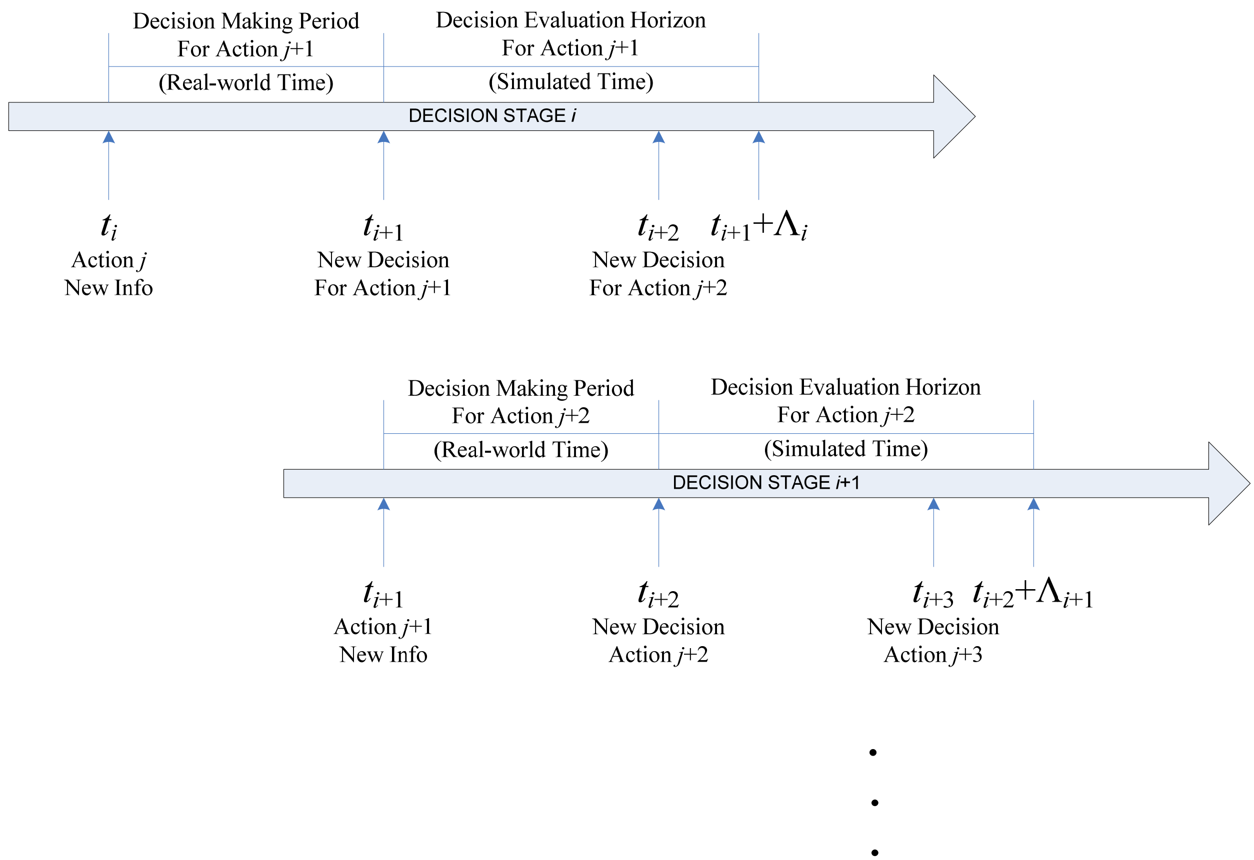

Figure 5-1. Evolutionary decision process

Algorithm [125]:

1. At the beginning of decision stage (iteration) *i*, time t_i:

    a. Deploy a new decision D_i which was made in the last iteration of the process. If $t = 0$ (i.e., start of the whole event), an expert or rule-based decision D_0 is preferred because there is no time allowed for running any computer programs to obtain solutions. (See example [82].)

    b. Real-time, actual data from the current time can be input into the system at this point as a new start.

2. Starting from time t_i, the simulation-based decision process is initiated. This process has to be finished by time t_{i+1} when the next decision stage starts. The real-world time period from t_i to t_{i+1} is called the decision making period or time between decisions. The decision made in the current iteration *i* will be deployed at time t_{i+1} and it should be proper (near-optimal) for a future period of time Λ_i from t_{i+1}. So this decision is evaluated by the simulation for the time period from t_{i+1} to $t_{i+1} + \Lambda_i$, which is called the decision evaluation horizon.

    a. With the new information input at time t_i, run the simulator to the next decision point t_{i+1} (simulated time). Store the simulation results of time t_{i+1} as SR_{i+1}. Since SR_{i+1} are the predicted results obtained by the simulator, they can be used as a part of the new information input in the next decision iteration *i*+1 starting from t_{i+1}.

    b. Construct an analytical model which can represent and replace the simulation model within certain error allowance; SR_{i+1} and/or earlier simulation results might be used to construct the analytical model. This model should be properly constructed for the new decision evaluation period from t_{i+1} to $t_{i+1} + \Lambda_i$. The length of the period Λ_i should be reasonably small because the analytical model has to use a considerable number of assumptions and simplifications and might only be valid over a limited range.

    c. Quickly solve the analytical model which can approximate the simulation system in order to promptly obtain a near-optimal solution $\widetilde{D}_{i+1}$ of t_{i+1}.

d. With the approximate near-optimal solution $\widetilde{D}_{i+1}$, conduct simulation-based local search to refine or improve the solution if possible, until the stopping criteria are reached. Record the best solution as D_{i+1}. In the local search process, the simulator runs from t_{i+1} to $t_{i+1} + \Lambda_i$ (simulated time) using SR_{i+1} as the starting data to evaluate the candidate solutions. The already explored solutions should be stored internally so that the simulation does not waste computation resources evaluating identical solutions more than once, although the chance of revisiting a solution is low. Normally, the stopping criteria are some predefined, computational conditions such as the length of computation time, the number of iterations, and/or the number of non-improvement iterations. In our case, since the decision making period (in Figure 5-1) is predefined, we choose to use the length of computation time as the optimization stopping criterion.

3. At the end of the current decision stage (iteration) *i*, time t_{i+1}:

   a. Deploy D_{i+1}, collect new information and cycle to the next stage *i*+1.

Figure 5-2 further illustrates the above evolutionary decision making procedure by dividing the iterations into four functional blocks/stages: preprocessing, analytical optimization, local search, and postprocessing. In the first preprocessing stage, the system is updated with new information/data and then one simulation is run to predict the situation at the future decision point. All the data collected from the new information and the simulation run will be used to prepare for the optimization procedures in the second stage. In the second analytical optimization stage, an analytical optimization model is constructed and solved quickly by some available optimization solvers (e.g., CPLEX [56], GLPK [46]). Its objective functions can approximately represent and replace the internal, inexplicit functions modeled by the simulation. Several constraints also need to be added to the program to maintain the feasibility of the solutions. This functional block provides a near-optimal solution to the system which can be used as a good starting point for the following local search process. In the third stage, a traditional local search is conducted in the neighborhood of the starting solution obtained previously. The neighborhood is constructed naively as the randomization of several solution components. After the stopping criteria are reached, the local search terminates, the final solution is output and new information/data is collected. This is the postprocessing stage.

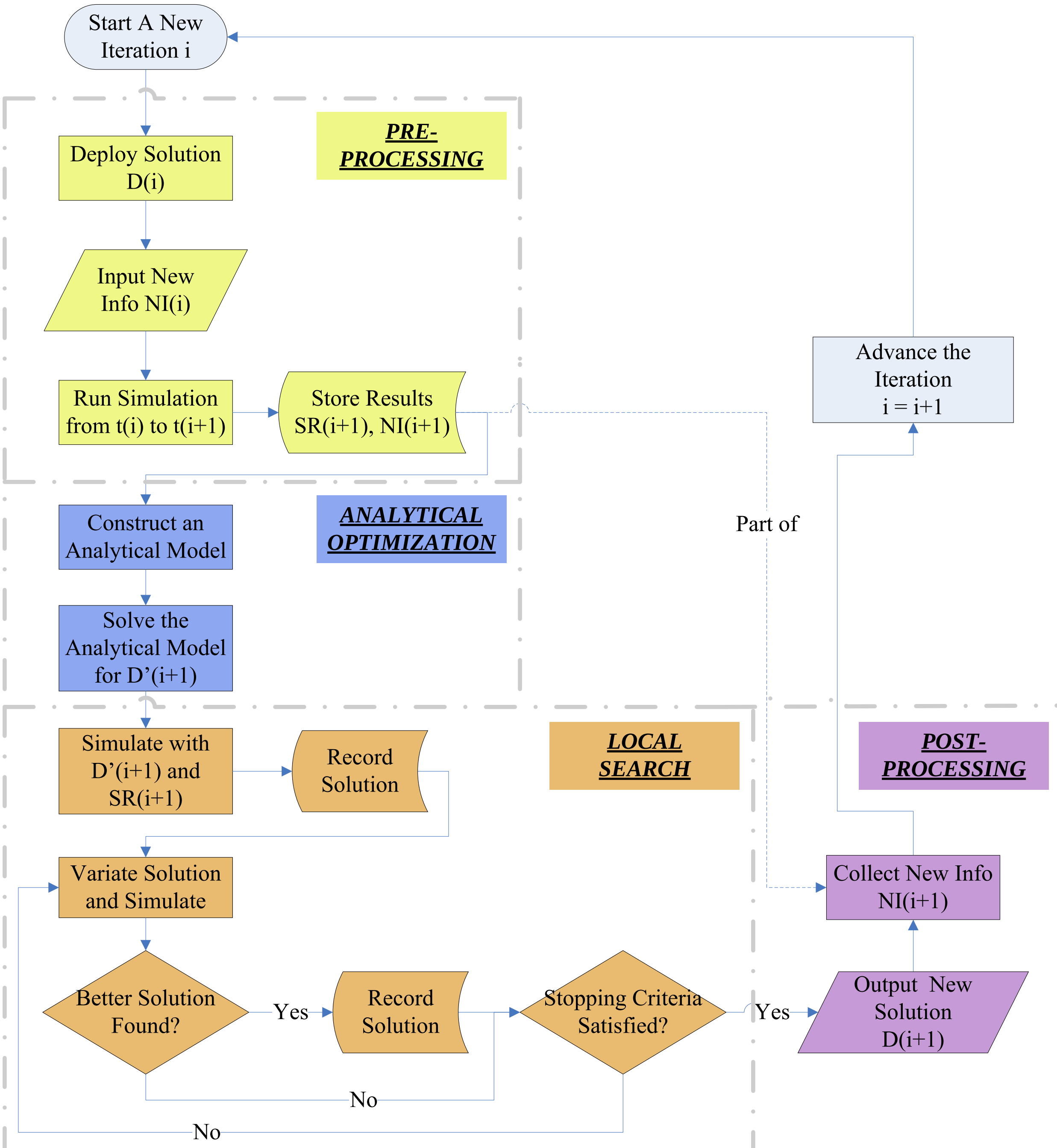

Figure 5-2. Evolutionary decision making flowchart (local search-based)

5.3 SELECTION OF TIME PARAMETERS

All events evolve with time, so the time parameters are the critical factors in the above evolutionary decision framework. This section elaborates on the discussion of those key heuristic parameters.

5.3.1 Decision making period

The selection of time points where we need to make/deploy new decisions (t_{i+1}, t_{i+2}, …) is itself a critical decision in the evolutionary decision making procedure. One may not want to wait too long to change the current decision as the system evolves and a new situation arises. On the other hand, if the new decision is too rushed, that decision may not be satisfactory because the situation is not perceived completely and the computational time is not enough for searching a good solution. Several ways for determining the length of the decision making period, i.e., $(t_{i+1} - t_i)$ for the i^{th} decision stage are described as follows:

- User-fixed decision making period. The advantage of fixing the decision making period is that it is more manageable and controllable for both the commanders and executants. On the management side, the time for simulation and optimization can be easily budgeted; on the execution side, they can be better prepared for the potential command changes in a fixed-time fashion. The drawback is that the users (i.e., decision makers who use the system) must have either good knowledge or estimate of the progress of the event otherwise the designated period may be too long or too short.
- Dynamic decision making period based on the event's progress. The decision point time is determined dynamically by the simulation results based upon the progress of the event,

e.g., when a specific measure changes by a predetermined amount. Such a period can be obtained in the following ways:

- Our simulator is capable of recording the situational progress in small time steps, say, 10 minutes of simulation time. Thus, with some explicit criteria, the decision making period can be easily obtained. Such period is dynamic because each run of the system is different from others as it changes with time.
- Projection methods can also be used to determine the decision period. Based upon the previous iterations of the process, the event's progress can be projected out to some extent. Even without running any simulation (which could be very expensive in terms of computation time), a decision period length can be estimated. However, this estimate can be inaccurate, especially in the early stages of the evolutionary decision process when few data are available. For example, at time 0 when no previous run is available, the only available information would be expert experience and historical data which may not be suitable for the specific case. Clearly, there is a tradeoff between the first method and this approach in terms of computational quality and time.
- Combine the above two methods. In the early stage of the process, use the first method which is time-consuming but accurate; after several iterations, use the second method which is time-efficient but must estimate based on previous run results. This combined approach may obtain both computational quality and efficiency.

- Dynamic decision making period based on special detection. A set of thresholds can be set up to detect when a new decision should be made. When some of the system parameters change significantly and suddenly or the event progresses unexpectedly, i.e., simulation results offset too much compared with reality, the threshold will be triggered and then a new decision process will be initiated. The advantage of this approach is that the system can compute only when necessary to save computational resources. The disadvantage is that most of the sudden system changes are not able to be predicted so the decisions should be made for a longer period of time which is hard or even invalid in general.

5.3.2 Decision evaluation horizon

The length of the decision evaluation horizon, e.g., Λ_i for decision stage *i*, is also an important parameter in the heuristic procedure as stated above. If the decision evaluation horizon is too long, the simulation runs will take too much time to evaluate and optimize the system. Additionally, the approximate analytical model may not estimate the simulation model well on a long time horizon. On the other hand, if the decision evaluation horizon is too short, i.e., the new decision is evaluated and optimized for a short time period, the solution obtained is myopic because the optimization process may neglect some further situations.

We hypothesize in general that for decision stage *i*, if the second future decision point t_{i+2} always resides in the decision evaluation horizon from t_{i+1} to $t_{i+1}+\Lambda_i$ for which the next decision is made, i.e., t_{i+2} is between t_{i+1} and $t_{i+1}+\Lambda_i$, the evolutionary decision process is expected to come up with better solutions than other options, given that the environmental situation does not change significantly in this decision stage. In other words, the decision evaluation horizon should be longer than the time between decisions, i.e., $\Lambda_i > t_{i+2} - t_{i+1}$, because a longer evaluation horizon allows the optimization process to gather more system response data and perceive the future situations better in order to achieve better solutions compared to the more near-sighted scheme. A better decision (if one exists) should be made before the system evolves to the point where the current solution reaches optimum for the previous decision. For instance, suppose new decisions are made on an hourly basis, the current time is 8:00 and we initialize a new decision stage. In this stage, we aim to make a new decision for 9:00 (denote the solution as $D_{9:00}$). To optimize $D_{9:00}$, the solution alternatives should be simulated and evaluated on a future decision evaluation horizon from 9:00 to sometime after 10:00, given that another near-optimal decision is made at 10:00. The rationale is that if a solution is optimal for the system in terms of the future objectives, the solution should drive the system to approach the optimal goal in every time step, otherwise the solution would not be the best candidate for the future. The hypothesis will be further tested and validated by some computational experiments in the later chapters.

Based upon this hypothesis, the time between the next two decisions $t_{i+2} - t_{i+1}$ should not be too long, because $\Lambda_i > t_{i+2} - t_{i+1}$ and the decision evaluation horizon Λ_i cannot be unnecessarily long as mentioned earlier in this subsection.

5.4 GENERAL METHODS FOR ANALYTICAL MODELING

The basic idea of the simulation-based optimization is to use simulation as the objective function evaluator to test various variable settings from which the relatively best solution then can be identified. Traditional simulation-based optimization algorithms need to explore a large solution space by using the simulation model to run a substantial number of solutions and search for the optimal or near-optimal one(s). For complex, large-scale simulation, running the model itself is extremely time-consuming. Especially when such large-scale simulation-based decision support systems are used in real-time management scenarios, the limited computation and decision time does not allow too many search trials as in the traditional algorithms. Quick identification of promising solution(s) is imperative. Analytical models are developed for this purpose in order to reduce the number of expensive simulation runs. They can be solved fast and represent the simulation model to some extent. Using the analytical results, only a few simulation runs should be required to heuristically search for a near-optimal or the optimal solution.

Although analytical models cannot totally replace the simulation, they can be used to approximate the simulation within an applied range and with some reasonable estimation errors. Several analytical modeling techniques are discussed and implemented in this dissertation. Although these techniques are problem dependent, certain modeling guidelines are universal as described below.

Direct mathematical modeling. For some system components, their internal operations can be expressed directly by closed-form mathematical equations, although some variations may be required. When a simulation model synthesizes numerous stochastic processes together and simulates their interactions, several subcomponents might have been well studied and the corresponding analytical formulations might have been derived. These well established results can be used directly to construct the analytical model. For instance, many subtle inventory analytical models have been developed in the past decades. In a global supply chain network simulation, a part of the entire system can be approximated by some analytical inventory models. Sometimes small efforts need to be taken to tailor the formulations to fit the specific problems. For example, Carley et al. [12] used a *Bioagent Delivered Dose* equation directly in the environmental submodel of the BioWar simulation.

Besides the existing formulations, some explicit mathematical functions are normally built into the simulation as a part of the model. These built-in functions can also be utilized to derive the analytical estimation model for the simulation.

The direct mathematical method may save considerable effort in the analytical modeling process but it does not work for many complex, non-classical systems due to the model constraints and/or the lack of existing research.

Linear regression. In many cases, statistical methods become more useful when the direct analytical relationships for complex systems are intangible or do not exist. Linear regression is a powerful statistical tool that can be used to determine the relationships between explanatory variables and response variables based on observed data. The simulation models take the explanatory variables as inputs and output the response values. The responses are used to determine the performance of the system. One advantage of system simulation is that numerous experiments can be conducted on the system in the preferred ways at much lower costs and risks than operating the real physical system. In order to use the linear regression method to construct analytical models, extensive simulation experiments need to be conducted to collect the response data as well as other variables and study the system behavior under different settings. For complex systems, multiple variables always present. A linear regression involving more than one explanatory variable is called multiple linear regression (MLR) [81]. The MLR can be conducted in the following steps:

1. Identify the response variable and corresponding decisive variables which might explain the shape of the response function. This is the first but the most important step because the selection of the variables determines the final functional form. To avoid overfitting the regression model later, there should not be too many explanatory (independent)

variables [62]. Each independent variable is expected to affect the response significantly. The regression significance will be addressed in the next steps. At this step, the knowledge and intuition to the model and physical meanings of the variables are always required.

2. Express the response variable using linear terms formed by the selected explanatory variables. At this stage, the first order linear terms of the explanatory variables always present because the variables are supposed to be able to impact the response significantly. Multiple order interaction terms might also have to be included. Some approaches to determine the multiple order terms are as follows:
   - Plot each of the multiple order terms (normally up to the third order) against the responses and investigate the impact. Incorporate those that appear to be "significant" in the initial model.
   - Analyze the physical meanings of the multiple order terms if possible. Incorporate the meaningful terms in the initial model.
   - Consult domain experts and incorporate the higher order terms validated by them based on their expertise.
   - Add the terms that cannot be decided whether important or not and let the significance test in the next step determine its existence.
3. Fit the model constructed in step 2 to the simulation results (observed data). Statistically test if the coefficients of the linear terms in the model are significant. If not, remove one most insignificant term at a time and refit and test until all the included terms' coefficients are significant. Besides the significance of coefficients, the goodness of fit of

a complete regression model can be evaluated by several other factors such as significance of regression and R^2 value.

4. Run more simulation experiments and further validate the above regression model by comparing the simulation results with the regression results. If the two match well, the analytical regression model is constructed successfully; otherwise, go back to step 1 and redo the whole process until a satisfactory model or another solution is obtained.

Although the multiple linear regression method is powerful in identifying the functional relationships, it is not suitable for modeling large-scale systems due to their complexity. The above procedure might only work well for modeling a component of the whole system. Thus, breaking a complex, intractable problem into small, tractable pieces to solve is a tactical concern here.

Expert modeling. If the analytical models are not obtained due to knowledge constraints, experts might be interviewed and/or existing literature could be searched and reviewed because their ideas may provide excellent insights into the problem. The experts' knowledge on the impacts of variables to responses is sometimes valuable to the analytical modeling process. Their expertise can be further validated by the simulation experiments. The evaluation of the system performance is another issue which needs experts' involvement. In many cases, empirical assessment functions are used to evaluate the system performance.

Combination method. None of the above methods may be satisfactory for the entire analytical modeling by itself. Combining two or more of them together is another good approach. For example, some of the parameters of the direct mathematical models can be obtained by linear regression; conversely, the linear regression model can be constructed partly by the direct mathematical models which bear physical meanings; experts' recommendations can always be incorporated in any of the models.

Analytical modeling is a time-consuming and demanding process, requiring a lot of engineering and mathematical skills. The development can be done offline while the final products (i.e., resulting models) are used online for real-time management scenarios. The analytical models are expected to represent the simulation system but their results are just estimates and are associated with errors. It is also worth noting that some of the parameters of the analytical models might depend on the simulation runs: they can hardly be determined beforehand. This modeling concept introduces the feedback system illustrated in Figure 5-3. In the closed-loop system, the previous simulation runs inform the analytical model of any changes in the situation and parameters; as the feedback, the analytical model can represent the simulation system more precisely so as to provide better solutions to the next simulation runs. Such an implementation improves the accuracy of the analytical model as the agent to the simulation.

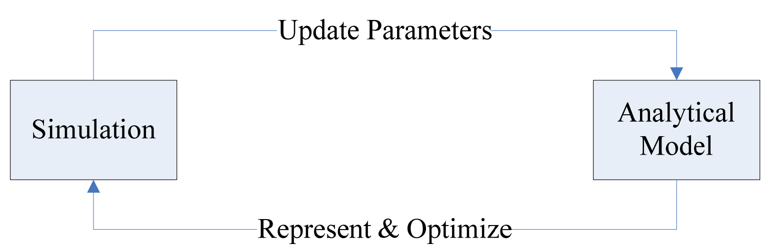

Figure 5-3. Simulation-analytics feedback system

5.5 ANALYTICALLY GUIDED RANDOMIZED SEARCH (AGRS)

Heuristic search is a common approach for simulation-based optimization. Almost all the simulation optimization algorithms consist of basic search procedures although numerous variations exist. The search algorithms start from a candidate solution and sequentially change the solution to search for a better one until the stopping criteria are satisfied. It is a local search if the new solution is in the neighborhood of the previous solutions. In the implementation of Figure 5-2, a classical local search procedure [45] is applied.

Local search often has difficulty reaching the global optimum because only a small neighborhood around the initial solution is explored. If the decision support system is used in real-time, only a few candidate solutions can be evaluated by an expensive (computer time) simulation model within a limited period of time. In this case, the optimality of local search largely depends on the quality of the initial solution. Storer et al. [109] developed a new method named problem space based local search that integrates fast, problem-specific heuristics with local search. The method basically perturbs the problem under study by adding a random vector of perturbations to the vector of problem data in order to generate a more effective neighborhood of solutions. The heuristic was implemented successfully for solving sequencing and scheduling problems.

In our algorithm, the local search's initial solution is given by an approximate analytical optimization model. The math model cannot exactly replicate the simulation due to such factors as simplifications, assumptions and linearization. However, the estimation errors of the analytical objective functions can be fitted to some statistical distribution with extensive data obtained from offline simulation experiments. By considering the analytical estimation errors, the previous evolutionary decision process can be modified as in Figure 5-4. An enhanced heuristic procedure takes the place of the traditional local search. We refer to this heuristic algorithm as *Analytically Guided Randomized Search* (AGRS) because it utilizes randomly perturbed analytical approximate solutions to guide the search. The details of the procedure are provided in the following discussion. Again, the already explored solutions should be stored internally throughout the procedure in order to save computation time when revisiting a solution although the probability of such an occurrence is not high.

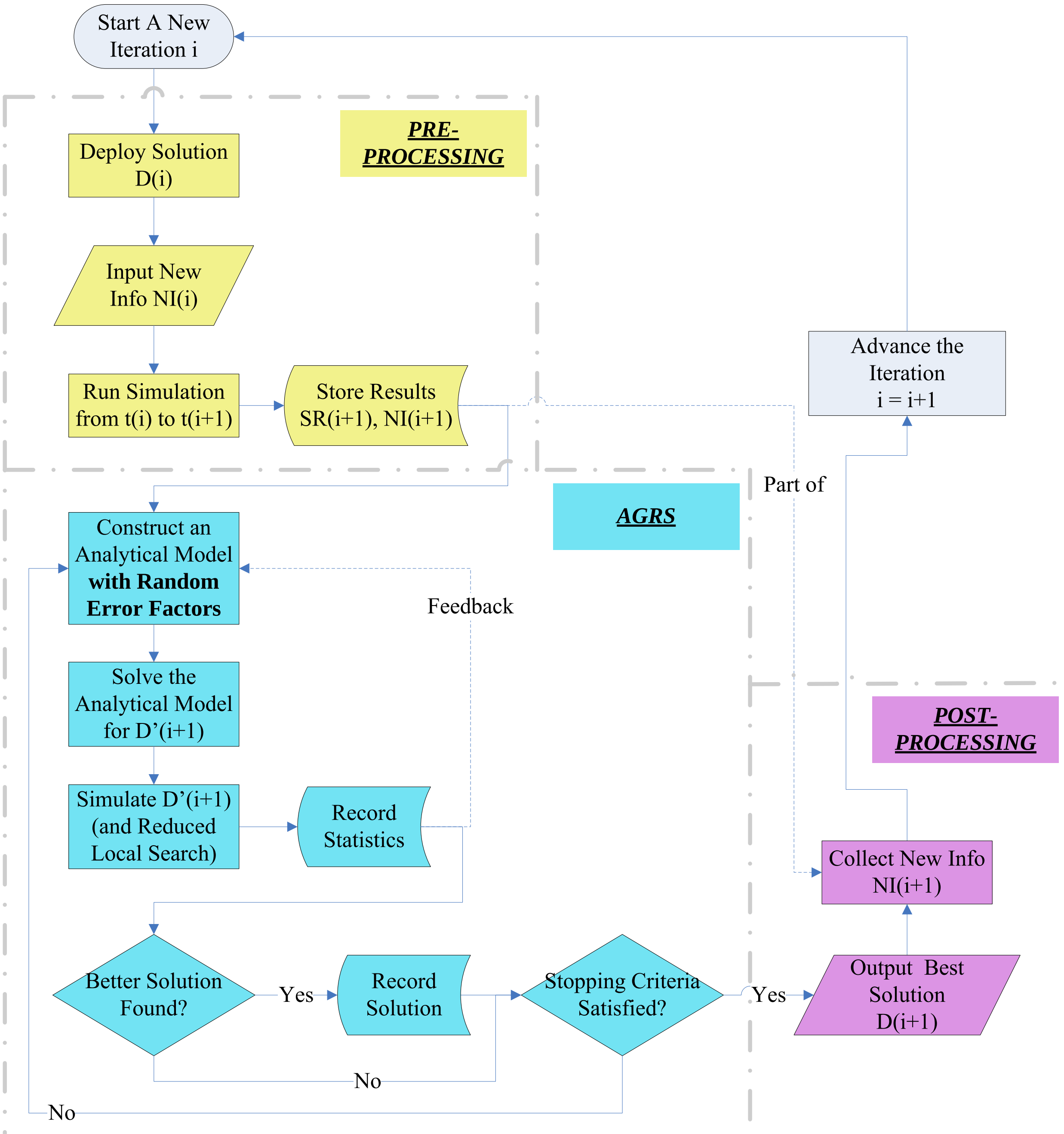

Figure 5-4. Evolutionary decision making flowchart (AGRS-based)

In each iteration of the AGRS procedure, an analytical model is built with the consideration of estimation errors. A modeling approach is suggested as follows:

$$f_e = e \cdot f_a \tag{5-1}$$

where the analytical function f_e is expressed as the multiplication of error factor e and pure, original analytical form f_a. The estimation error can be modeled directly as a random distribution obtained from offline experiments or some functional form that incorporates certain distributions. In the AGRS-based evolutionary decision process, the local search iterations can be reduced or even eliminated to save computational time for more analytically guided randomized search iterations. The AGRS procedure has more advantages than the classical local search discussed previously (the computational comparison is available later in subsection 6.3.3.3). This process enhances solution. It enables us to search among various rational solutions given by perturbed analytical models instead of conducting naïve local search blindly around only one initial solution. Furthermore, AGRS can avoid being trapped in a local optimum when the pure analytical model is associated with relatively large errors, which is true especially for large-scale, complex models.

5.6 APPLICATION FIELD OF THE PROCEDURE

Generality is one of the most critical performance measures for simulation optimization algorithms. A well-constructed optimization procedure is expected to deal with a wide variety of problems of interest [33]. The simulation-based evolutionary real-time decision procedure developed in this dissertation can be applied in different areas to solve many complex problems which have the following properties:

- Long process time. The system should evolve for a significant long period of time.

- Sequential information input. Partially observed or inaccurate information is available in the beginning, but it improves with time.
- Optimal streams of decisions. Different decisions are needed to adapt to different situations.

In this dissertation, we use agent-based discrete event simulation as the main tool to model the disaster response system. The integrated simulation system not only involves the responders' operations and actions but also includes complex interactions such as victim deterioration, weather impact and traffic congestion. One great advantage of our disaster responder simulation system – D^4S^2 – is the seamless integration with other components including a geographic information system (GIS), user-friendly graphical interfaces and disaster information databases [121, 122, 123]. The users can interact with the system to a very high level. With such great flexibility, better (i.e., more accurate and more complete) information is streamed gradually into the system and helps to improve the quality of the simulation results and the decision made in the online use of the system. While the disaster advances, the situation may change and new decisions are needed to better manage the overall responses. Thus, disaster event management is a great example to apply the simulation-based evolutionary decision procedure.

Besides disaster planning, global supply chain network, financial system portfolio management and military base management could be other excellent examples to apply the procedure. All have the common properties described in the beginning of this subsection.

5.7 SUMMARY

A validated simulation system is a potentially valuable tool for evaluation of system performance. When simulation marries optimization, it turns out to be a powerful tool for making decisions. This chapter develops a novel simulation-based evolutionary decision making procedure, or metaheuristic. The basic idea of the metaheuristic algorithm is to divide the entire time horizon into smaller intervals and optimize the solutions for these individual intervals, allowing for the dynamics between intervals. The prior interval solutions may affect the later ones.

The simulation optimization algorithm utilizes approximate analytical optimal solutions to guide the heuristic search into promising (near optimal) regions quickly followed by a few local search iterations. An improved version of the heuristic aims at promptly identifying multiple such promising regions by perturbing the coefficients of the analytical model and solving the model.

The evolutionary decision making procedure can be applied to broad classes of large-scale, complex problems which involve long process time, sequential information input, and optimal streams of decisions. Some example applications are disaster planning, global supply chain network, financial system portfolio management and military base management.

The next chapter applies the methods described here to the disaster response management problem specifically. The computational results demonstrate the usefulness and effectiveness of the system.

6.0 DISASTER SIMULATION DECISION SYSTEM

This chapter utilizes the disaster response simulation model implemented in Chapters 3.0 and 4.0 to provide a concrete, sophisticated example of the application of the simulation-based evolutionary decision making procedure developed in the last chapter. The decisions focus on the logistical dispatching issues for first responders and secondary responders.

In civilian disaster management, the decisions pertain mainly to allocation of emergency resources, and the dispatch of responders to control the affected population and assets. The decision makers want to determine the best response times and number of the necessary responders sent to the disaster scene and obtain the best overall outcomes by executing such decisions. Some of the factors that might significantly impact the outcomes are: (1) resource allocation such as the number of each resource to be sent off and (2) resource dispatch sequence which is the time-sequence for dispatching various responders. For the resource allocation management, sufficient but not redundant resources should be sent to the scene. Redundant resources do not help but can increase congestion and slow other responders' missions. Space-consuming vehicles like fire trucks should be dispatched prudently since they are congestion makers in narrow downtown areas. On the other hand, adequate resources must be allocated to the disaster response to meet the minimum requirements and accomplish the mission. For the resource sequencing, different resources' dispatching times have to be determined according to their purpose and priority. Besides the disaster response, the coverage of the normal emergency events is also considered in the decision system; otherwise, more fatalities might result than from the major disaster event itself.

6.1 BASIC ANALYTICAL MODEL

The disaster response simulation system can model the actual system more realistically than analytical models but it requires a considerable amount of computer time to execute. The disaster response decisions are extremely urgent and must be made in a limited time frame. Therefore, using the time-consuming simulation runs to search for the optimal (or near-optimal) solutions becomes impractical. Thus, an analytical nonlinear mixed-integer program (NMIP) is developed in this research to remedy the intractable simulation-based heuristic search. The NMIP model can be solved rapidly compared to simulation runs so as to guide the search into a promising solution space very quickly. With the high-quality (near-optimal) initial solutions given by the NMIP, only a few full simulation runs are needed to obtain good-quality solutions. The modeling details are described in the following subsections.

6.1.1 Nonlinear mixed-integer programming formulation

Multiple objectives are involved typically in complex decision problems and part or all of the objectives are competing against each other. For example, in our case, we want to dispatch more emergency vehicles to the scene in order to increase the victim evacuation capacity but too many vehicles would introduce significant congestion and negatively impact the whole efficacy eventually. In multi-objective problems, the Pareto optimal solutions are always preferred [22].

A solution is efficient or Pareto optimal if there are no feasible solutions that are at least as good in every objective. The set of Pareto optimal solutions is called the efficient frontier or the tradeoff curve. A solution is dominated if there is another solution better in one objective and at least as good in the rest. One approach to find Pareto-optimal points is called the weighted sum method which combines the objectives with some weights. If all weights are positive, the combined single-objective program would give a Pareto optimal point, if an optimal solution exists. The weights can normally be decided by the expert users after evaluating the relative importance of all the objectives.

First of all, we formulate the multi-objective nonlinear mixed-integer program by closely investigating the internal structure and operations of the simulation model. The model is designed to have seven main objectives and one auxiliary objective as listed below [125]. Those are all important measures for the disaster response management. Note that all the objectives are evaluated for the fixed-length time period of Λ starting from t_{i+1} in decision stage i, as defined in Figure 5-1.

Objective 1 (Q_1): Maximize number of life-threatening victims evacuated.

Objective 2 (Q_2): Maximize number of severe victims evacuated.

Objective 3 (Q_3): Maximize number of moderate victims evacuated.

Objective 4 (Q_4): Minimize number of fatalities at the scene.

Objective 5 (Q_5): Minimize EMS normal incident response degradation.

Objective 6 (Q_6): Minimize penalty cost for calling mutual aid agents.

Objective 7 (Q_7): Minimize penalty cost for changing tasks (communication cost).

Objective 8 (Q_8): Minimize dispatching distance.

It is believed that longer out-of-hospital time will increase mortality although the existing medicine literature does not fully validate/support this correlation [69] because the studies are all retrospective, i.e., the studies were done after the emergency events and they might not represent the situations during the events [66]. Trunkey [117] stated three distinct periods in which fatally injured trauma victims will die: (i) immediate, within the first “Golden Hour”, (ii) early, 2-3 hours after injury, and (iii) late, several days to several weeks after injury. The study showed that about half of all deaths occur in the immediate period and 30% in the early period. Thus, the out-of-hospital time, including response time and time to hospital, is an important measure in emergency medicine. Also, for a mass casualty disaster, the scene can be better controlled and managed if the victims can be evacuated more quickly. The first three objectives are to maximize the victim evacuation number within the time period Λ. It is equivalent to minimizing the total out-of-hospital time for a population.

Without adequate medical treatment and stabilization, the life-threatening victims can deteriorate to deaths fast. The human lives are the first priority for the responders to consider so the number of fatalities at the scene, as will be modeled in the fourth objective, is another important objective to minimize besides the victim evacuation time.

The fifth objective is to balance the normal emergency responses and the disaster response. We cannot afford a major loss of normal calls during a disaster event.

The sixth and seventh objectives consider some cost terms for the response task assignment. According to the mutual aids agreement (e.g., [85]), the out-of-area responders can be called to assist in responding to the major disaster event but in this case the coverage of their home areas will be lost to some extent. Therefore, such a loss should be penalized by adding a cost term. In the developed evolutionary decision making framework, the response decisions are evaluated and adjusted in every time interval. But the response assignments cannot be changed too frequently because the responders need to take time and efforts to communicate and switch their roles. This incurs another cost term for extra communications. The cost functions are normally not as important as the first five objectives in the disaster response scenarios.

In order to respond to the disaster more quickly, we always want to dispatch the responders who are closer to the disaster scene. The last objective serves this purpose.

Emergency response planning is basically a nonlinear assignment problem. Emergency vehicles (e.g., ambulances) are modeled as the agents in the agent-based simulation model. Each agent is assigned one response task. The literature related to nonlinear assignment programs can be found in [18, 90].

Let N be the set of all n available emergency vehicle agents. They are divided into subsets S_1, S_2, …, S_r according to their types. In other words, type i agents are included in subset S_i such that $S_i \subseteq N$, $\bigcup_i S_i = N$, and $S_i \bigcap S_j = \phi$, $\forall i \neq j \in \{1,2,...,r\}$. In the specific example, three most important and common types of agents are considered. They are advanced life support (ALS) ambulances, basic life support (BLS) ambulances and fire trucks (first responders).

$S_1 = \{\text{ALS ambulances}\}$,

$S_2 = \{\text{BLS ambulances}\}$ and

$S_3 = \{\text{Fire trucks (first responders)}\}$.

One of four possible tasks needs to be assigned to each of the agents. The four tasks are mutually exclusive and collectively exhaustive; they are:

Task 1: Evacuating disaster victims.

Task 2: Staying at the disaster scene and stabilizing victims.

Task 3: Responding to normal emergency incidents in the disaster area.

Task 4: For mutual aid units, responding to home area.

Besides the objective functions, several constraints are needed to ensure the feasibility of the solutions. The basic NMIP model is presented below.

$$\underset{X}{\text{Minimize}} \quad \sum_j w_j \cdot Q_j(X) \tag{6-1}$$

Subject to

$$\sum_k x_{ik} = 1, \quad \forall i \in N, \tag{6-2}$$

$$\sum_{i \in S_1 \cup S_2} x_{i1} \geq 1, \tag{6-3}$$

$$\sum_{i \in S_1 \cup S_2} x_{i3} \geq 1, \tag{6-4}$$

$$x_{i4} = 0, \quad \forall i \notin S_{\text{MutAid}}, \tag{6-5}$$

$$x_{ik} = \begin{cases} 1, & \text{if agent } i \text{ is assigned to task } k; \\ 0, & \text{otherwise.} \end{cases} \tag{6-6}$$

The objective function (6-1) aggregates several individual objectives Q_j by imposing positive weights w_j to each of them. Without loss of generality, we minimize the aggregate objective function. If any individual objective Q_j needs to be maximized, the Q_j's sense should be changed in order to keep its weight w_j positive. As mentioned before, the objective weights can be determined by consulting domain experts (e.g., emergency managers, commanders) and evaluating the multiple objectives' relative importance. We should assign higher weights to more important objectives in order to favor them over the minor objectives. However, different objectives can have different meanings, units and/or ranges, so it could be challenging to decide their relative importance. A reasonable setting of the objective weights is given in the following. The first three objectives are all related to the number of people evacuated from the scene and the fourth objective is about the number of deaths at the scene. It is not hard to determine their relative importance because they have the same physical meaning: people's headcount. It is more important to reduce the number of deaths at the scene (the fourth objective) than to evacuate more victims. Among the first three objectives, evacuating life-threatening patients has the most importance because they can decay to deaths quickly; while evacuating moderate patients is minor. Therefore, for the first four objectives, their relative weights should be in the order: $w_4 > w_1 > w_2 > w_3$. The fifth objective is related to the EMS normal call response degradation which has a totally different physical meaning than the first four objectives. Furthermore, their values are on different scales. The response degradation is a decimal value less than 1.0 while the number of patients (headcount) is essentially an integer value which can range from dozens to hundreds. Since the response degradation is on a lower scale, we want to assign a higher weight to it in order to allow it to compete with other objectives. Therefore, for the first five objectives, their relative weights should be in the order: $w_5 > w_4 > w_1 > w_2 > w_3$. The last three objectives are about mutual aid cost, changing task cost and dispatching distance, respectively, which are relatively minor considerations for the disaster response management, so lower weights should be assigned to them. It is worth noting that the above relationship of the objective weights is just one suggested setting and it can be changed for different situations and different management needs.

The decision variables x_{ij} are binary as defined in (6-6). They indicate the response assignment for each vehicle agent. The task responses are mutually exclusive and collectively exhaustive so the integrity constraint (6-2) is necessary. Constraint (6-3) regulates that at least one EMS unit, either ALS or BLS vehicle, should be assigned to respond to the major disaster event for victim evacuation. Constraint (6-4) assumes that at least one EMS unit should respond to the normal emergency events, because otherwise, the basic coverage of the city will be lost totally. However, this constraint can be relaxed. Only the medical responses are considered for the normal events in this model. For a major disaster event, the city command can call for the assistance from nearby resources according to specific mutual aid agreements. Task 4 is only designed for out-of-area mutual aid vehicles. If they are called, some costs are incurred because the service in their original service areas will be downgraded. If they are committed to their original service area, no cost is incurred. Constraint (6-5) says that only the mutual aid agents could respond to their original service areas, where S_{MutAid} is the agent subset of all mutual aid vehicles.

The optimization program's constraints have been written explicitly with binary variables. Those linear constraints are relatively straightforward compared with the objective functions. In the later sections, nonlinear objective functions Q_j's will be described in detail.

6.1.2 Problem complexity and tractability

The optimization problem defined above is an assignment problem with general nonlinear objective functions. Since the quadratic assignment problem (QAP) is known to be NP-hard [18, 36, 90], the general nonlinear assignment problem must also be NP-hard. Furthermore, in simulation optimization, each candidate solution can only be evaluated by simulation and this can be computationally demanding for large-scale models.

Heuristic search is one of the well established approaches to obtain good solutions in a fast manner for NP-hard problems. To design a local search algorithm, finding a suitable neighborhood is the critical first step. A simple neighborhood relation of this problem would be the following: *L*' is a neighbor of *L* if *L*' can be obtained by the change of one agent's task assignment. The size of this neighborhood is less than or equal to $(t-1)\times n=\theta(n)$, where t the number of possible tasks, n is the totally number of agents that are assigned tasks. This is not a large neighborhood but evaluating all the combinations by the simulator is not practical given that one run takes about 15 minutes[2], and the search has to be done and a good-quality solution has to be obtained within a limited time frame, say, one hour.

If a small neighborhood is chosen, the computational effort of checking neighbors will be relatively small but the search could be easily trapped in a local optimum; on the other hand, if a large neighborhood is allowed, checking neighbors will become very expensive but the algorithm has more capability to shun the local optima. One major concern of simulation studies is the run time, especially when it is applied to such time-limited decision making problems as real-time disaster management. In this case, a quick choice of a good starting point for the heuristic search can save significant time for decisions.

Although the traditional local search algorithms will not work perfectly for the limited-run simulation optimization, its working principles can be of great help in solving the problem. The key idea is to start from a good solution, check its local vicinity, and move along the direction from the good solution to a better one. The search proceeds iteratively until reaching a satisfactory solution. This idea is implemented in our evolutionary real-time decision making

---

[2] The computing environment is referred to subsection 4.4.1.

procedure. A quickly solvable model is paramount in this research. In the next section, analytical estimated objective functions for the program presented in subsection 6.1.1 will be formulated.

6.2 ANALYTICAL OBJECTIVE ESTIMATION

The mathematical program formulated above in subsection 6.1.1 is an assignment problem with nonlinear objectives. This section details the construction of these nonlinear objective functions using the general methods presented in section 5.4. Linearization techniques are then applied to reformulate the mathematical program in order to solve it using standard state-of-the-art MIP solvers, e.g., CPLEX [56].

6.2.1 Nonlinear mixed-integer programming formulation

In this subsection, we want to model the eight objective functions (discussed in section 6.1) in explicit mathematical forms one by one. Some of these functions incorporate the previous simulation results but can be solved by efficient solvers independently of the expensive simulator. Those functions only provide estimates of the real, unknown objective functions with errors, however we have to sacrifice some quality in order to reduce computational efforts. A modeling tradeoff is involved in this context. The objective functions are formulated for the decision evaluation horizon from t_{i+1} to $t_{i+1}+\Lambda_i$ in a specific decision stage i (refer to Figure 5-1). A fixed length Λ is used for the horizon in the following formulations. The objective functions incorporate many nonlinear factors, integer and continuous decision variables such that the entire model turns out to be a nonlinear mixed-integer program.

6.2.1.1 Maximize number of life-threatening victims evacuated

Due to the chaos and/or shortage of medical personnel and equipment at the disaster scene, injured victims should be evacuated from the scene to fully functional medical facilities, e.g., hospitals, as quickly as possible. The following three objectives (i.e., 1, 2 and 3) are to maximize the evacuation number of life-threatening, severe and moderate types of patients, respectively.

EMS ambulances (both ALS and BLS types) can have two major functions in the disaster responses: stabilization and evacuation. For stabilization, they can transport medical responders along with medical supplies to the scene and let the responders stay at the scene to treat and stabilize the casualties. The on-scene stabilization can help reduce mortality and it will be considered in the later subsection 6.2.1.4. The first three objectives primarily deal with victim evacuation. For evacuation, ambulances can continuously stabilize and transport the casualties to hospitals. Different types of ambulances, i.e., ALS vs. BLS, have different transportation capacities. For example, according to the transportation triage rules defined in subsection 4.2.3, ALS can transport three patients at most: one life-threatening and one severe patients in the back cabinet and one moderate victim on the front seat; while BLS can only take a moderate patient due to its limited space and lower skill level in medical treatment.

Ambulances travel back and forth between the scene and hospitals to evacuate victims. So the total evacuation capacity within a time period can be estimated by a simple travel time-speed relationship as follows:

$$\mathrm{EvacCap} = \frac{\mathrm{TimePeriod}}{\mathrm{EvacRate}} \tag{6-7}$$

The evacuation rate in the denominator of equation (6-7) is the rate at which one victim is transported to hospital. The ambulances' evacuation task consists of four segments: treating and loading patients at the scene, traveling to the hospital, unloading patients at the hospital, and traveling back to the scene [121]. The at scene activities can be significantly retarded by the scene congestion which is determined by many factors such as the scene infrastructure condition and number of emergency vehicles involved at the scene. Thus, the standard at-the-scene time may be adjusted by some congestion factor function (as discussed in subsection 4.3.2). The next three segments of the evacuation task are combined together and their total time is called "Trip Time" because they represent a whole round-trip from scene to hospital and back to scene. Note that one ambulance may take more than one victim, which is determined by its capacity. The evacuation rate can be expresses as:

$$\text{EvacRate} = \frac{\text{TripTime} + (\text{SceneTime} + \text{CongestionTime})}{\text{VehicleCap} \cdot \text{VehicleNum}} \tag{6-8}$$

Combining (6-7) and (6-8), we get:

$$\text{EvacCap} = \text{VehicleCap} \cdot \text{VehicleNum} \times \frac{\text{TimePeriod}}{\text{TripTime} + (\text{SceneTime} + \text{CongestionTime})} \tag{6-9}$$

Considering that both ALS and BLS may be involved in the evacuation, we can build the evacuation capacity function in two parts separately (i.e., ALS evacuation + BLS evacuation) as follows:

$$z_{lt} = c_{1-lt} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + (t_{1-load} + t_a)} + c_{2-lt} \cdot \sum_{i \in S_2} x_{i1} \times \frac{\Lambda}{\tau_{2-trip} + (t_{2-load} + t_a)} \tag{6-10}$$

where,

z_{lt} : Total number of life-threatening patients evacuated during the time period Λ.

c_{1-lt} : Type 1 vehicle (ALS)'s life-threatening capacity per trip.

c_{2-lt} : Type 2 vehicle (BLS)'s life-threatening capacity per trip.

Λ : The length of decision evaluation horizon.

τ_{1-trip} : Type 1 vehicle (ALS)'s round-trip time excluding load time at the scene.

τ_{2-trip} : Type 2 vehicle (BLS)'s round-trip time excluding load time at the scene.

t_{1-load} : Type 1 vehicle (ALS)'s load time at the scene.

t_{2-load} : Type 2 vehicle (BLS)'s load time at the scene.

t_a : A responder vehicle's time to access the scene with congestion.

Note that the Greek-letter notations represent the parameters which are obtained from previous simulation runs. For instance, Λ depends on the event progress which is evaluated by the simulation model; τ_{1-trip} is the type 1 vehicle (ALS)'s round-trip time which can be estimated by the previous runs of the simulation.

The scene congestion factor is modeled in the scene accessing time t_a as stated in subsection 4.3.2; t_a is a nonlinear function in terms of the total space occupied by the response vehicles: $\sum_i s_i(x_{i1} + x_{i2})$ where s_i is the vehicle congestion factor which is related to the vehicle's dimension (for example, fire truck = 2.0, ALS ambulance = 1.5, and BLS ambulance = 1). A responder vehicle's time to access the scene with congestion is formulated as follows:

$$t_a = t_s \left(\frac{c}{c - \sum_i s_i(x_{i1} + x_{i2})} \right)^{\alpha} \tag{6-11}$$

where t_s is the vehicle's standard time (without congestion) to access the disaster scene, c is the scene space capacity at which the scene will be fully congested without any possible traffic flow, s_i is the vehicle congestion factor, and α is a power parameter.

Let z_{lt} in formula (6-10) be the estimated evacuation capacity for life-threatening patients in the time period Λ. The initial number of patients at the scene may or may not be larger than the estimated capacity. The total number of patients that can be evacuated should be expressed as:

$$Q_1 = \min\{z_{lt}, \lambda_2\} \tag{6-12}$$

where λ_2 is the initial number of life-threatening patients at the scene for the current time interval Λ. To write the objective function (6-12) in a valid mathematical program, we want to introduce a new variable $z_{lt}^{(ex)}$ which represents the amount of excessive capacity for the life-threatening patients. The excessive life-threatening evacuation capacity can be consumed by other less urgent (e.g., severe) victims, so the variable $z_{lt}^{(ex)}$ is utilized in the next objective function to model the evacuation of severe patients. Since the evacuation capacity can never be negative, $z_{lt}^{(ex)}$ should be modeled in a piece-wise form:

$$z_{lt}^{(ex)} = \begin{cases} 0 & \text{if } z_{lt} \le \lambda_2 \\ z_{lt} - \lambda_2 & \text{otherwise} \end{cases} \tag{6-13}$$

To implement the above piece-wise function, we want to use a binary variable $y_2^{(ex)}$ to flag whether the evacuation capacity is excessive or not and add the following constraints:

$$\begin{aligned} z_{lt} - \lambda_2 &\le M y_2^{(ex)} \\ \lambda_2 - z_{lt} &\le M\left(1 - y_2^{(ex)}\right) \end{aligned} \tag{6-14}$$

where M is a "sufficiently" large constant multiplier to ensure that the above constraints always hold. With the constraint set, we can rewrite function (6-13) as follows:

$$z_{lt}^{(ex)} = y_2^{(ex)} \cdot \left(z_{lt} - \lambda_2\right) \tag{6-15}$$

By integrating the above formulations, we can obtain the sub-program for the first objective, i.e., to maximize the number of life-threatening patients evacuated from the scene, as:

$$
\begin{aligned}
\max \quad & Q_1 = z_{lt} - z_{lt}^{(ex)} \\
\text{s.t.} \quad & z_{lt} = c_{1-lt} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + \left(t_{1-load} + t_a\right)} + c_{2-lt} \cdot \sum_{i \in S_2} x_{i1} \times \frac{\Lambda}{\tau_{2-trip} + \left(t_{2-load} + t_a\right)} \\
& t_a = t_s \left(\frac{c}{c - \sum_i s_i \left(x_{i1} + x_{i2}\right)} \right)^{\alpha} \\
& z_{lt}^{(ex)} = y_2^{(ex)} \cdot \left(z_{lt} - \lambda_2\right) \\
& z_{lt} - \lambda_2 \le M y_2^{(ex)} \\
& \lambda_2 - z_{lt} \le M\left(1 - y_2^{(ex)}\right) \\
& y_2^{(ex)} \text{ binary}, x_{i1} \text{ binary}
\end{aligned}
\tag{6-16}
$$

Since all objectives will be aggregated into a minimization (chosen arbitrarily) objective, the above maximization objective is manipulated to:

$$
\min \quad Q_1 = -z_{lt} + z_{lt}^{(ex)} \tag{6-17}
$$

The 2nd and 3rd objectives can be built in the similar manner as follows.

6.2.1.2 Maximize number of severe victims evacuated

The sub-program for this objective is similar to (6-16) except that the leftover evacuation capacity from the upper level, i.e., life-threatening, patients is added to the total evacuation capacity for this level, i.e., severe patients.

$$
\begin{aligned}
\min \quad & Q_2 = -z_{svr} + z_{svr}^{(ex)} \\
\text{s.t.} \quad & z_{svr} = c_{1-svr} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + \left(t_{1-load} + t_a\right)} + c_{2-svr} \cdot \sum_{i \in S_2} x_{i1} \times \frac{\Lambda}{\tau_{2-trip} + \left(t_{2-load} + t_a\right)} + z_{lt}^{(ex)} \\
& z_{svr}^{(ex)} = y_3^{(ex)} \cdot \left(z_{svr} - \lambda_3\right) \\
& z_{svr} - \lambda_3 \le M y_3^{(ex)} \\
& \lambda_3 - z_{svr} \le M\left(1 - y_3^{(ex)}\right) \\
& y_3^{(ex)} \text{ binary}, x_{i1} \text{ binary}
\end{aligned}
\tag{6-18}
$$

where,

z_{svr} : Total number of severe patients evacuated during the time period Λ.

$z_{svr}^{(ex)}$: The amount of excessive capacity for severe patients' evacuation.

$y_3^{(ex)}$: Flag for whether the capacity for severe patients' evacuation is excessive or not.

c_{1-svr}: Type 1 vehicle (ALS)'s severe capacity per trip.

c_{2-svr}: Type 2 vehicle (BLS)'s severe capacity per trip.

λ_3: The initial number of severe patients at the scene for the current time interval.

6.2.1.3 Maximize number of moderate victims evacuated

Just as the previous objectives, this objective's sub-program can be written as follows:

$$\begin{aligned}
\text{min} \quad & Q_3 = -z_{mm} + z_{mm}^{(ex)} \\
\text{s.t.} \quad & z_{mm} = c_{1-mm} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + \left(t_{1-load} + t_a\right)} + c_{2-mm} \cdot \sum_{i \in S_2} x_{i1} \times \frac{\Lambda}{\tau_{2-trip} + \left(t_{2-load} + t_a\right)} + z_{svr}^{(ex)} \\
& z_{mm}^{(ex)} = y_4^{(ex)} \cdot \left(z_{mm} - \lambda_4\right) \\
& z_{mm} - \lambda_4 \le M y_4^{(ex)} \\
& \lambda_4 - z_{mm} \le M\left(1 - y_4^{(ex)}\right) \\
& y_4^{(ex)} \text{ binary}, x_{i1} \text{ binary}
\end{aligned} \tag{6-19}$$

where,

z_{mm}: Total number of moderate/minor patients evacuated during the time period Λ.

$z_{mm}^{(ex)}$: The amount of excessive capacity for moderate/minor patients' evacuation.

$y_4^{(ex)}$: Flag for whether the capacity for moderate/minor patients' evacuation is excessive or not.

c_{1-mm}: Type 1 vehicle (ALS)'s moderate/minor capacity per trip.

c_{2-mm}: Type 2 vehicle (BLS)'s moderate/minor capacity per trip.

λ_4: The initial number of moderate/minor patients at the scene for the current interval.

6.2.1.4 Minimize number of fatalities at the scene

Without sufficient medical care, the condition of the victims at the scene could deteriorate fast especially for the high-risk population such as the life-threatening victims. They will die at a relatively high rate and the rate increases with the elapsed time. In this problem, we only consider the deterioration from life-threatening to death so here in this subsection we use word "patients" or "victims" to represent the life-threatening cases only. In most response situations, it is necessary to dispatch some pre-trained medical personnel to the scene who can assist in stabilizing the patients' conditions.

In subsection 4.3.1, two approaches to modeling the scene victim degradation are presented: closed form model and agent-based simulation model. In this subsection, the corresponding analytical models will be developed. They are called closed-form victim degradation model and simulated victim degradation model, respectively.

A. Closed-form victim degradation model

The theoretical foundation and implementation of closed-form victim degradation for the disaster simulation was discussed in subsection 4.3.1. In the simulator, we look all the living patients as a group and keep their total number in a counter to reduce the computational overhead. The whole time period Λ is uniformly divided into a sequence of time intervals with constant length δ_t (e.g., five minutes, simulation time). So the total number of small intervals n_{int} is:

$$n_{\text{int}} = \Lambda / \delta_t \tag{6-20}$$

For the (n-1)$^{\text{th}}$ small simulation time step, the number of added fatalities is calculated by:

$$x_{n-1} = v_{n-1} p \tag{6-21}$$

where v_{n-1} is the total number of living patients in the beginning of the (n-1)th time interval and p is the proportion of victims who will decay (i.e., decease) during this time interval. Note that p does not change over time and it can be obtained from the formula (4-8). Transform equation (6-21) and we can calculate the total number of patients in the beginning of the (n-1)th time interval as:

$$v_{n-1} = \frac{x_{n-1}}{p} \tag{6-22}$$

Then, the total number of patients who will survive (i.e., remain at the scene) during the (n-1)th time interval w_{n-1} is expressed as:

$$w_{n-1} = \frac{x_{n-1}}{p}(1-p) \tag{6-23}$$

where $(1-p)$ is the proportion of patients who will survive in the interval. Besides deterioration, the living patients at the scene are evacuated by the transportation responders such as EMS ambulances. Assuming the victim evacuation rate is constant over the entire time period Λ, we can calculate the number of patients who are evacuated during the time interval (denoted as n_e) by:

$$n_e = \frac{\left(z_{lt} - z_{lt}^{(ex)}\right)}{n_{\text{int}}} = \left(z_{lt} - z_{lt}^{(ex)}\right)\delta_t / \Lambda \tag{6-24}$$

where $\left(z_{lt} - z_{lt}^{(ex)}\right)$ is the total number of life-threatening patients evacuated from the scene during time period Λ and it is formulated in subsection 6.2.1.1. Then we can deduct n_e from the total number of survived patients w_{n-1} to obtain the number of patients who remain at the scene at the end of the (n-1)th time interval, i.e., in the beginning of the nth time interval, denoted as v_n.

$$v_n = \frac{x_{n-1}}{p}(1-p) - n_e \tag{6-25}$$

Knowing the proportion of deaths p is fixed for every interval, we can write out an iteration equation about the fatalities at the scene as follows:

$$x_n = v_n p = \left(\frac{x_{n-1}}{p}(1-p) - n_e\right)p \tag{6-26}$$

where x_n is the number of new deaths generated in the n^{th} interval and x_{n-1} is the number of fatalities in the immediately previous time interval.

By simplifying the above formula, we obtain:

$$x_n = (1-p)x_{n-1} - n_e p \tag{6-27}$$

We then iterate the formula further and use mathematical induction to express x_n by the first term x_1 as follows:

$$\begin{aligned}
x_n &= (1-p)x_{n-1} - n_e p \\
&= (1-p)^2 x_{n-2} - n_e p(1+(1-p)) \\
&= (1-p)^3 x_{n-3} - n_e p\left(1+(1-p)+(1-p)^2\right) \\
&= \ldots\ldots \\
&= (1-p)^{n-1} x_1 - n_e p\left(1+(1-p)+(1-p)^2 + \ldots\ldots + (1-p)^{n-2}\right)
\end{aligned} \tag{6-28}$$

The last part of the above expression forms a finite geometric series with the ratio of $(1-p)$. We can further simplify the formula by summing up the geometric series.

$$\begin{aligned}
x_n &= (1-p)^{n-1} x_1 - n_e p \frac{1-(1-p)^{n-1}}{1-(1-p)} \\
&= (x_1 + n_e)(1-p)^{n-1} - n_e
\end{aligned} \tag{6-29}$$

By plugging $n = 1$ into the above equation, we can obtain $x_1 = x_1$. This verifies the correctness of the equation. x_1 is the number of fatalities in the first interval. It equals $\lambda_2 p$ where λ_2 is the initial number of life-threatening patients at the scene.

Again, the right-hand side of the above equation forms a finite geometric series with the ratio of (1 – p). We can calculate the total number of fatalities for the whole time period Λ by summing up the fatalities in each time interval as follows:

$$\sum_{n=1}^{n_{\text{int}}} x_n = (x_1 + n_e)\frac{1-(1-p)^{n_{\text{int}}}}{p} - n_e n_{\text{int}} \tag{6-30}$$

where n_{int} is the number of small time intervals divided in the period Λ and it is calculated by:

$$n_{\text{int}} = \Lambda / \delta_t \tag{6-31}$$

By integrating all the related equations above, we can obtain the objective function of total number of fatalities at the scene during the time period Λ as:

$$\begin{aligned} \min \quad & Q_4 = (\lambda_2 p + n_e)\frac{1-(1-p)^{\Lambda/\delta_t}}{p} - n_e \Lambda / \delta_t \\ \text{s.t.} \quad & n_e = \left(z_{lt} - z_{lt}^{(ex)}\right)\cdot \delta_t / \Lambda \\ & p = 1 - g^{-\lambda\cdot\delta_t} \\ & \lambda = a^{-\left(\sum_i w_i x_{i2}\right)} \end{aligned} \tag{6-32}$$

where w_i represents the relative skill level of the individual responders in terms of medical treatment.

B. <u>Simulated victim degradation model</u>

In this model, the life-threatening patients are modeled as individual agents in the simulator (see subsection 4.3.1.2). Although it is a bit more computationally expensive than the closed-form model, this approach can consider more complex interactions among the agents and produce more realistic, accurate results to the actual system. For the closed form model, mathematical formula can be used directly and explicitly in the simulator and optimization program as was done in the previous subsection. For the simulated victim degradation model, its system behavior has to be studied statistically through extensive computer experiments before being applied to the analytical optimization program. A small agent-based discrete event simulation model is built specifically for studying such behavior as medical responders' efficacy for on-scene patient stabilization, evacuation responders' efficacy for on-scene fatality reduction. This victim degradation sub-model is one component of the whole simulator but can be run independently.

In subsection 4.4.2, we have shown the positive impact of medical responders on the patients' condition through a set of simulation experiments. See the results in Figure 6-1. Without any medical treatment, the patient's condition will deteriorate most quickly along the blue solid-diamond curve. If more responders who have higher capacity for providing medical treatment are on the scene, the corresponding decay curve will be offset lower which means that more lives can be saved in a specified amount of time. The efficacy of medical responders is defined as the total number of lives that can be saved by some point in time, compared with no any medical treatment (i.e., natural decay). For example, the gap denoted by the arrow in Figure 6-1 is the efficacy of one medical responder by 3.5 hours from the beginning.

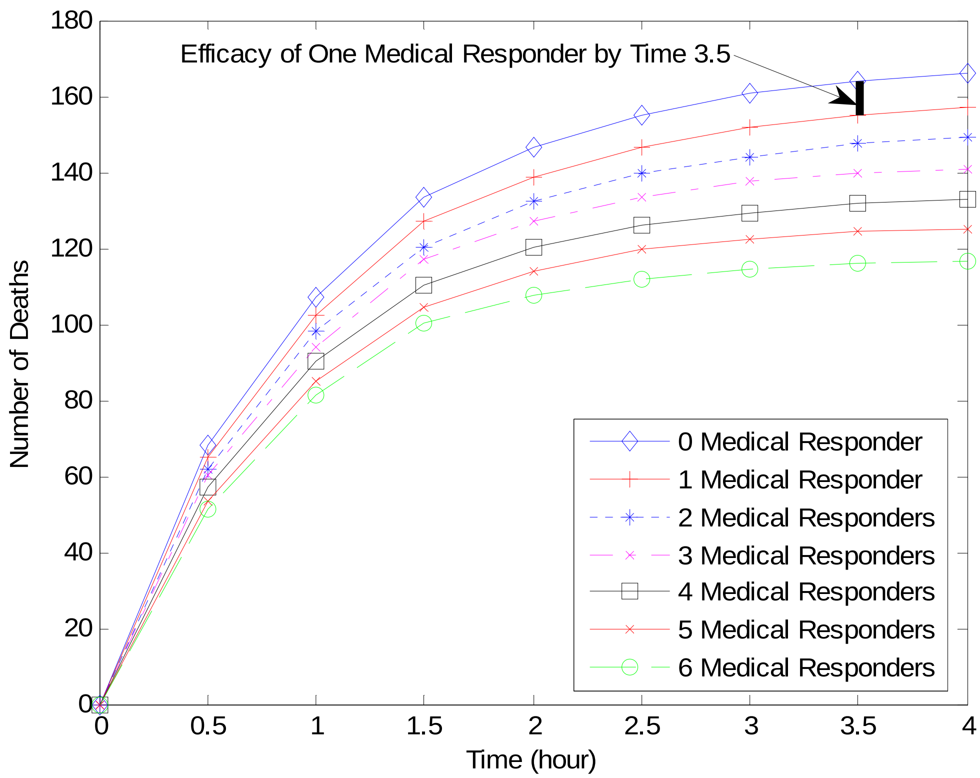

Figure 6-1. Impact of on-scene medical responders to victim degradation

As stated in subsection 4.4.2, the efficacy of medical responders is proportional to the number of them. Define unit efficacy as the number of lives that can be saved by only one responder by time t. We have shown the number of fatalities can be modeled by

$$f(t) = n(t) - a \cdot u(t) \cdot m \tag{6-33}$$

where n is the number of natural death without any medical care, a is a constant coefficient (approximately equals to 1.0 in this case as modeled in subsection 4.4.2), u is the unit efficacy of one medical responder and m is the total number of medical responders. We call this the one-dimensional casualty prediction model because it assumes all the medical responders are identical and have the same efficacy for treating the patients. Due to the continuous deterioration, n and u as well as f increase monotonously over time. Given the number of initial casualties and their decay property (i.e., decay base parameter defined in equation (4-9)), we can quickly run a simulation to obtain the values for n and u at every time step. With this information and the prediction model (6-33), we can estimate the number of fatalities at the desired point of future time.

The goodness of fit of the above prediction model is tested through an intensive random experiment. In each run of the experiment, three inputs: number of initial casualties, decay base (defined in subsection 4.3.1) and number of medical responders are randomly generated within certain ranges. Some actual simulation results and the mathematical predictions to simulation by the model (6-33) are compared in Figure 6-2. The model performs extremely well such that almost all predictions are within 10% error relative to the actual simulation results. In statistics, if fitting the one-dimensional casualty prediction model to the simulation result data, the p-value of the F test turns out to be less than 0.001 which means the fit is significant. Further, the corresponding R^2 value is 0.997 which indicates an extremely high goodness of fit level. We can conclude that they are statistically consistent with each other so the one-dimensional casualty prediction model is valid.

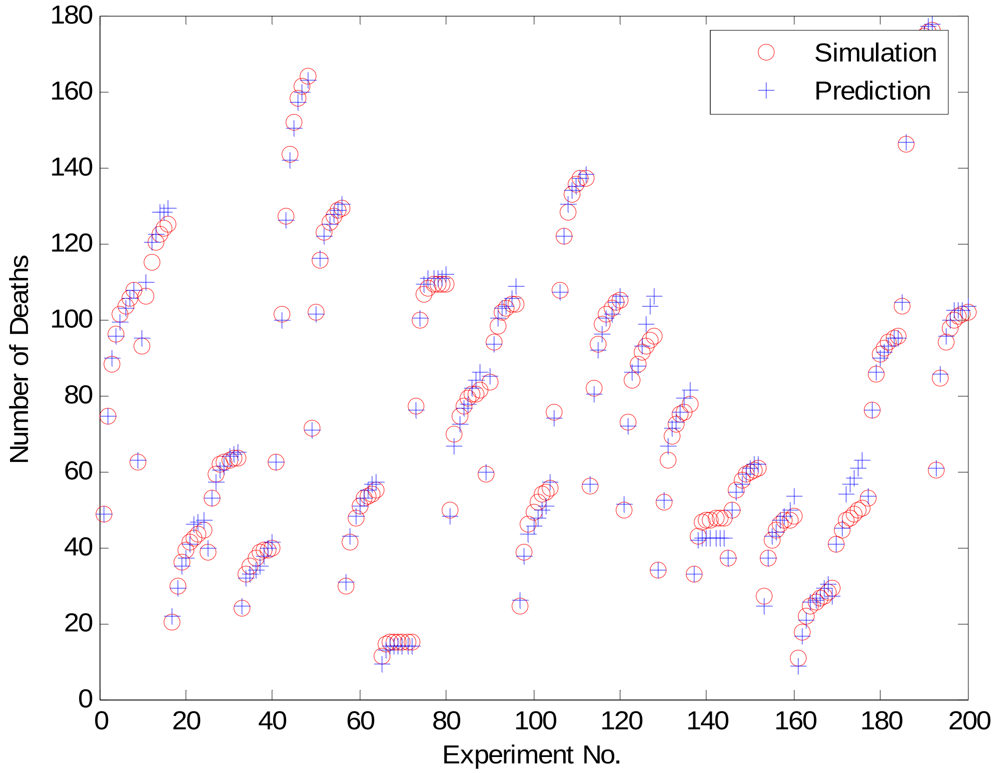

Figure 6-2. Experiment on one-dimensional fatality prediction model

Different responders have different skill levels in terms of patient treatment and stabilization. The differences are reflected in the victim survival decay function after treatment. When a hybrid team of medical personnel work together to treat a group of patients, they follow a set of predefined rules (see examples in Chapter 4.0). Based upon the one-dimensional casualty prediction model (6-33) and a pilot simulation experiment on the hybrid-team treatment, we hypothesize that the efficacy of a group of different types of medical responders is related to the individual efficacy of each responder and the total number of responders, and the following mathematical prediction model is proposed:

$$f(t) = n(t) - \left(a_0 + \sum_i a_i \cdot u_i(t) \cdot m_i + \sum_{i,j \geq i} b_{ij} \cdot u_i(t) u_j(t) \cdot m_i m_j \right) \tag{6-34}$$

The notation is the same as in (6-33) except that the indices *i*'s and *j*'s are used to differentiate the types of medical responders and b_{ij} is the second-order constant coefficient. There are three types of medical personnel included in this study: ALS, BLS and first responders so *i* and *j* range from 1 to 3. We call this the multi-dimensional fatality prediction model because multiple types of responders are involved and they have different responding skill levels. The above formulation is essentially a second-order model and it considers both individual medical responder's efficacy and the effects of their interactions.

First of all, we can quickly run a small one-dimensional victim degradation simulation to obtain the values for $n(t)$ and $u(t)$ so they are regarded as known parameters. Then, by running extensive random experiments, we collect about 1000 sample data points and then use linear regression to obtain the coefficients *a*'s and *b*'s for model (6-34). The initial regression result is shown in Table 6-1.

Table 6-1. Regression result for multi-dimensional fatality prediction model (1st attempt)

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.975
R Square	0.951
Adjusted R Square	0.951
Standard Error	5.267
Observations	1004.000

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	9.000	535630.606	59514.512	2145.344	0.000
Residual	994.000	27574.797	27.741		
Total	1003.000	563205.402			

Table 6-1 (continued)

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	0.958	0.677	1.416	0.157	-0.370	2.287	-0.370	2.287
M1	1.193	0.034	35.420	0.000	1.127	1.259	1.127	1.259
M2	1.109	0.035	31.568	0.000	1.040	1.178	1.040	1.178
M3	0.980	0.045	21.550	0.000	0.891	1.069	0.891	1.069
M1^2	-0.004	0.000	-8.536	0.000	-0.005	-0.003	-0.005	-0.003
M2^2	-0.003	0.001	-5.695	0.000	-0.004	-0.002	-0.004	-0.002
M3^2	0.000	0.001	-0.519	0.604	-0.002	0.001	-0.002	0.001
M1*m2	-0.002	0.001	-2.884	0.004	-0.004	-0.001	-0.004	-0.001
M1*m3	-0.005	0.001	-5.598	0.000	-0.006	-0.003	-0.006	-0.003
M2*m3	0.000	0.001	-0.458	0.647	-0.003	0.002	-0.003	0.002

In the first attempt, we have included all the possible first-order and second-order terms in the above regression model. But this model is not valid because not all of the regression terms are significant to the regression model. The iterative multiple regression process with backward elimination scheme [81] is then performed. According to the *p*-values evaluated at significance level 0.05, all the coefficients of first-order terms (i.e., a's excluding a_0 in model (6-34)) have significant contribution to the dependent variable f. For the same reason, the second-order terms m_1^2, m_2^2, m_1*m_2 and m_1*m_3 are significant contributors. Other coefficients are not statistically significant so we attempt to eliminate them one at a time from the model and determine a revised model. The final regression result is shown in Table 6-2.

Table 6-2. Regression result for multi-dimensional fatality prediction model (final)

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.975
R Square	0.951
Adjusted R Square	0.951
Standard Error	5.263
Observations	1004.000

Table 6-2 (continued)

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	7.000	535615.417	76516.488	2762.249	0.000
Residual	996.000	27589.985	27.701		
Total	1003.000	563205.402			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	1.158	0.620	1.868	0.062	-0.059	2.376	-0.059	2.376
M1	1.195	0.033	35.745	0.000	1.129	1.261	1.129	1.261
M2	1.104	0.033	32.975	0.000	1.038	1.170	1.038	1.170
M3	0.953	0.022	42.345	0.000	0.909	0.997	0.909	0.997
M1^2	-0.004	0.000	-8.570	0.000	-0.005	-0.003	-0.005	-0.003
M2^2	-0.003	0.001	-5.787	0.000	-0.004	-0.002	-0.004	-0.002
M1*m2	-0.002	0.001	-3.039	0.002	-0.004	-0.001	-0.004	-0.001
M1*m3	-0.005	0.001	-5.709	0.000	-0.006	-0.003	-0.006	-0.003

In the final regression result, all the coefficients of single and cross-term variables are shown to be significant at level 0.05. We can also conclude that the regression is significant and has a good prediction capability due to the low *p*-value for the F test and high R^2 value, respectively. In the above regression process, for each run of the experiments, we randomized the input parameters such as the number of initial casualties, survival function bases before and after treatment and the number of medical responders, so the resultant regression model covers a wide variety of scenarios within certain reasonable ranges.

The goodness of fit of the model is verified through another random simulation experiment. In contrast to the experiment conducted for the one-dimensional fatality prediction model, this experiment incorporates three types of medical responders with different efficacy levels. We randomly generate the number of responders of each type. An example of predicted and actual simulation results are plotted and compared in Figure 6-3. Because multiple types of responders and their interactions are considered, the multi-dimensional fatality prediction model is more complex in nature. As a result, it is not as good as the one-dimensional model with regard to the prediction quality. But the model has adequate estimation capability to be used in the analytical optimization model. The mean and median of the relative error of predicted values to actual simulation results are 27.74% and 8.40% respectively. A Wilcoxon signed-rank test is performed to compare the simulation and prediction sampling results. At the 95% confidence level, the confidence interval for the median of the difference of simulated and predicted results is [-0.558, 0.186] which contains 0. So we can conclude that the two models (simulation vs. prediction) are not significantly different from each other and the mathematical prediction model is valid.

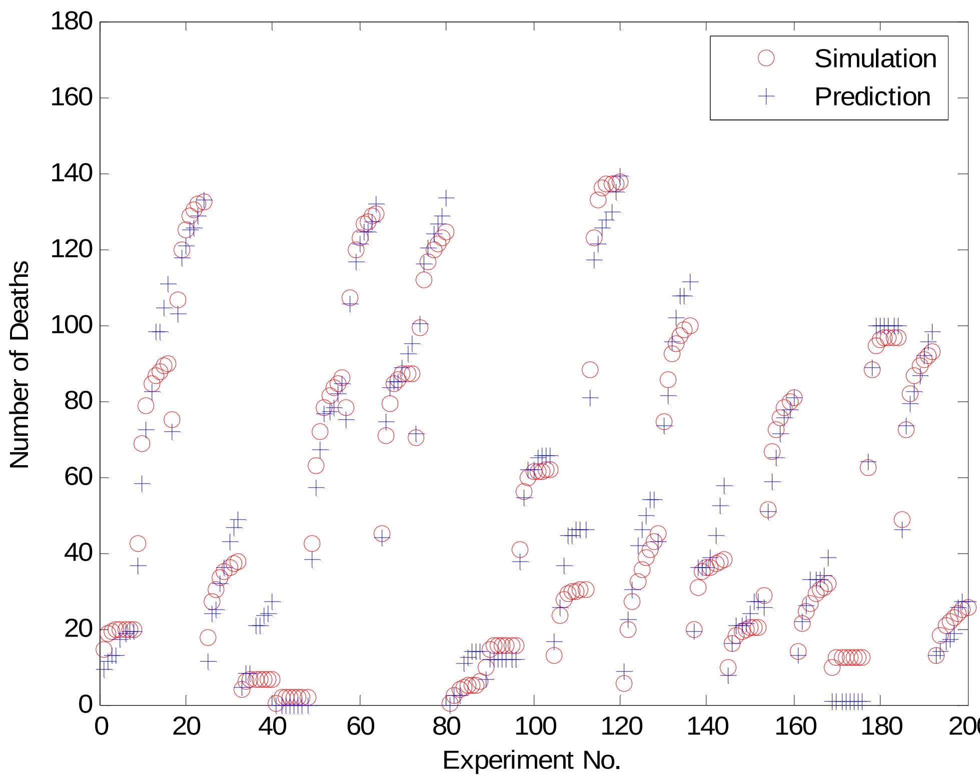

Figure 6-3. Experiment on multi-dimensional fatality prediction model

Besides on-scene treatment and stabilization, the disaster victims are evacuated at a certain rate which is related to other factors such as vehicle evacuation capacity and traveling speed. With the rules and assumptions stated in Chapter 4.0 that the evacuation ambulances always transport the most critical patient from the scene, the total number of fatalities should be reduced by the number evacuated. Using the format of the above multi-dimensional fatality prediction model, we have

$$f(t) = n(t) - \left(a_0 + \sum_i a_i \cdot u_i(t) \cdot m_i + \sum_{i,j \geq i} b_{ij} \cdot u_i(t) u_j(t) \cdot m_i m_j \right) - g(t) \tag{6-35}$$

where g is the fractional capacity of evacuation within a period of time and it can be expressed as follows.

$$g(t) = Q_1 \cdot t / \Lambda \tag{6-36}$$

The life-threatening evacuation number Q_1 is modeled in subsection 6.2.1.1.

Equation (6-35) is a more comprehensive scene fatality prediction model with the consideration of different types of scene stabilization and evacuation capabilities. Since the total number of fatalities is accumulated, the f function should increase monotonically with time. The following constraint should be imposed to all of the above related formulations to obtain more reasonable estimations.

$$f(t_{j+1}) \geq f(t_j) \tag{6-37}$$

Figure 6-4 compares some analytical prediction results and simulation results from a set of random experiments. It shows that the two set of results are close to each other so the analytical prediction model has adequate predictive capability. The mean and median of the relative error of predicted values to actual simulation results are 19.51% and 5.33% respectively. A Wilcoxon signed-rank test is performed to compare the simulation and prediction sampling results. At the 95% confidence level, the confidence interval for the median of the difference of simulated and predicted results is [0.00, 1.03] which contains 0. So we can conclude that the two models (simulation vs. prediction) are not significantly different from each other and the complete mathematical prediction model is valid.

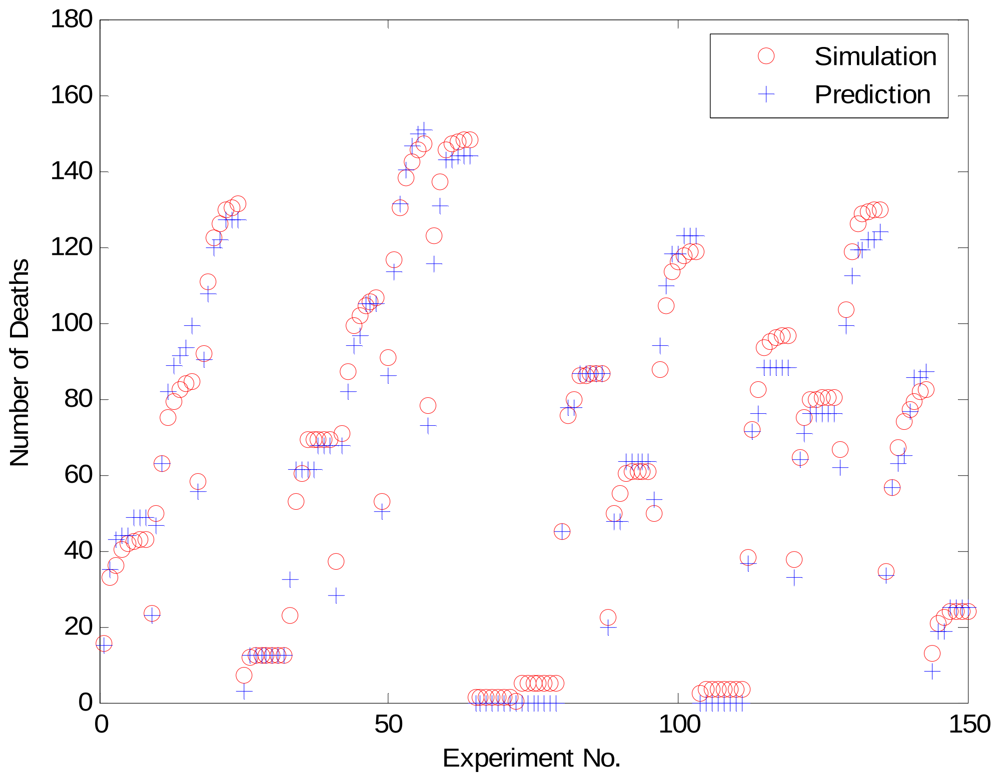

Figure 6-4. Experiment on the complete fatality prediction model

The regression model presented above can be used directly in modeling the scene fatalities Q_4. It is a function of the number of responders. However, the function parameters (i.e., u's in the above models) are not directly available as constants. Whenever the model is used, those parameters can be obtained by quickly running a component of the whole simulation. This introduces a new modeling approach – component modeling and optimization.

A comparison of the closed-form and the simulated victim degradation models is given here to conclude this subsection. The closed form model incorporates available mathematical formulations directly in the simulation, so such equations can be easily manipulated into the analytical optimization program. However, the analytical equations used may not represent the actual system as well as the simulation model does. The simulated victim degradation model enables the victim agents to interact with the responder agents and mimic the situation more realistically. However, the simulation model does not have explicit analytical forms so it is hard to be formulated into the optimization program. Although statistical regression can be applied to obtain some analytical relationships for the simulation, such relationships could have large errors and they are not as robust as the closed-form equations. Hence, a modeling tradeoff is involved here, and both model complexity and quality have to be considered. The closed-form victim degradation model is chosen in this dissertation for the following computational experiments.

6.2.1.5 Minimize EMS normal incident response degradation

While responding to a major disaster, the emergency management team should also consider the coverage for the normally occurring emergencies in the service area. The performance of responders is mainly related to the number of in-service EMS vehicles. Intuitively, more vehicles should result in better response coverage if they are appropriately located. There are also other stochastic factors that affect the system performance such as the locations and volume of emergency calls. Normal response rules are built into the simulator. We run the simulation experiments with different numbers of normal responding vehicles, and then fit a regression model to capture the relationship between response performance and number of responding agents.

One standard indicator of EMS system performance is the response time (RT) interval between call receipt and arrival on scene. A study by Blackwell and Kaufman [6] was conducted in a metropolitan county – Mecklenburg County, North Carolina – on a single-tier, paramedic service EMS system. Through the observational study, a conclusion was drawn that the survival rate can be improved when the emergency calls' RTs are less than five minutes; when the RTs exceed five minutes, such improvement is minor. Shuman et al. [105] used eight minutes as a break point to category satisfactory and unsatisfactory responses in RURALSIM – a simulation system for evaluating both rural and urban EMS services. For different deployments of the system, this criterion may vary. In this research, the normal incident response performance is evaluated by the degradation level. It is defined as the proportion of the emergency calls that are responded longer than a certain amount of time, e.g., eight minutes used in Shuman et al.'s work. Simulation experiments are run to compare the normal call response degradation levels for different number of responding agents. In the set of experiments, normal emergency calls are generated randomly in the network based on certain distributions and then EMS vehicles are simulated to respond to those normal calls according to predefined response rules. For a given number of EMS responders, several random experiments are performed and the average normal call response degradation value is calculated. By varying the number of EMS responders and repeating the experiments, we can obtain a typical relationship between normal call response degradation and the number of responders as shown in Figure 6-5 where results are marked by blue dots and are fitted by red piecewise lines. The trend is that more responding EMS vehicles result in a lower normal call response degradation level because more resources can respond to the events more efficiently and more calls are satisfied within the response time standard.

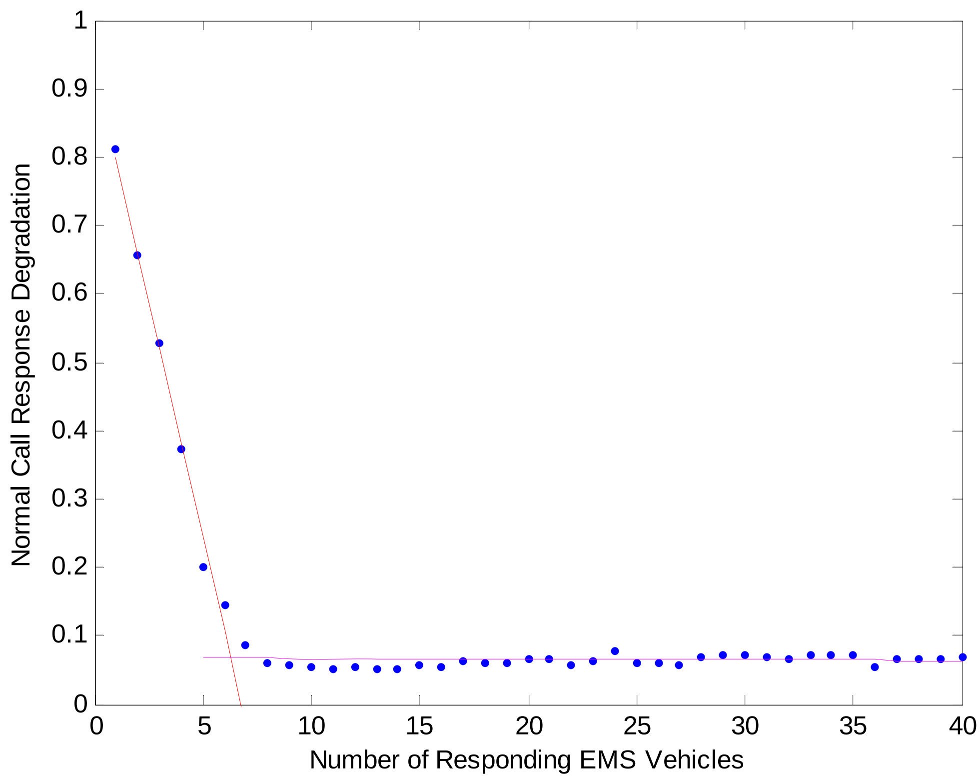

Figure 6-5. Normal call response degradation curve

The above simulation results (marked by blue dots) can be fitted by a piecewise linear curve (n pieces) with negative slopes. So the program can be formulated as:

Minimize $\quad z \qquad$ (6-38)

Subject to

$$z \geq a_1 \cdot \sum_{i \in S_1 \cup S_2} x_{i3} + b_1 \qquad (6\text{-}39)$$

$$z \geq a_2 \cdot \sum_{i \in S_1 \cup S_2} x_{i3} + b_2 \qquad (6\text{-}40)$$

$$\ldots$$

$$z \geq a_n \cdot \sum_{i \in S_1 \cup S_2} x_{i3} + b_n \tag{6-41}$$

where a_1, a_2, …, a_n are the slopes for the n pieces of lines, respectively; and b_1, b_2, …, b_n are the intercepts for the n pieces of lines, respectively. S_1 and S_2 corresponds to the sets of ALS and BLS vehicles, respectively.

6.2.1.6 Minimize penalty cost for calling mutual aid agents

During a major large-scale disaster event, the city's emergency management team may want to call for assistance from neighboring areas according to the predefined mutual aid agreements (e.g., [85]). However, calling mutual aid will cause other serious problems such as destroying the service balance and leaving some areas uncovered. From the above regression experiment result shown in Figure 6-5, the response performance is approximately linearly related to the number of in-service vehicles within some specified range, so we can also evaluate the penalty cost for calling mutual aid vehicles linearly in terms of the number of called responders. Such cost should be minimized:

$$\text{Minimize} \qquad \sum_h w_h \cdot \sum_{k=1,2,3} \sum_i \pi_{ih} x_{ik} \tag{6-42}$$

where,

π_{ih} : A 0-1 matrix to flag if agent i belongs to the mutual aid area h originally.

w_h : Mutual aids penalty weight for the mutual aid area h.

x_{ik} : Response task assignments, $x_{ik} = \begin{cases} 1, & \text{if agent } i \text{ is assigned to task } k; \\ 0, & \text{otherwise.} \end{cases}$.

6.2.1.7 Minimize penalty cost for changing task assignments

For ease of operations and logistics during the disaster event, the task assignment for a particular agent should not be changed frequently during the disaster event, if possible. Switching the roles of the agents will incur some cost, e.g., raising the complexity of communications. We want to minimize such cost by adding the following objective.

$$\text{Minimize} \qquad \sum_{i,k}\left(x_{ik} - \widetilde{x}_{ik}\right)^2 \tag{6-43}$$

where $\widetilde{x}_{ik}$'s are the previous response task assignment solution. The quadratic term forces the function to be positive. It is worth noting that other reasonable forms of this objective function, e.g., $\sum_{i,k}\left|x_{ik} - \widetilde{x}_{ik}\right|$, are also possible.

6.2.1.8 Minimize dispatching distance

The agents dispatched to the disaster scene should be the ones that are closer in time to the scene location in order to respond more promptly. To meet this requirement, the last objective is added. This objective functions only when there is a tie among various solution alternatives and it does not compete with other objectives so its objective weight should be much smaller than others.

$$\text{Minimize} \qquad \sum_{i} d_i\left(x_{i1} + x_{i2}\right) \tag{6-44}$$

where d_i is the i^{th} agent's distance to the scene in the beginning of the simulation and d_i 's can be obtained by real-time information input or estimated from previous responses.

To summarize, the complete aggregate nonlinear mixed-integer program is as follows:

$$\underset{X}{\text{Minimize}} \quad \sum_j w_j \cdot Q_j(X)$$

Where,

$$Q_1 = -z_{lt} + z_{lt}^{(ex)}$$

$$Q_2 = -z_{svr} + z_{svr}^{(ex)}$$

$$Q_3 = -z_{mm} + z_{mm}^{(ex)}$$

$$Q_4 = (\lambda_2 p + n_e)\frac{1-(1-p)^{\Lambda/\delta_t}}{p} - n_e \Lambda/\delta_t$$

$$Q_5 = z$$

$$Q_6 = \sum_h w_h^{(ma)} \cdot \sum_{k=1,2,3} \sum_i \pi_{ih} x_{ik}$$

$$Q_7 = \sum_{i,k} (x_{ik} - \widetilde{x}_{ik})^2$$

$$Q_8 = \sum_i d_i (x_{i1} + x_{i2})$$

Subject to

$$\sum_k x_{ik} = 1 \quad \forall i \in N$$

$$\sum_{i \in S_1 \cup S_2} x_{i1} \geq 1$$

$$\sum_{i \in S_1 \cup S_2} x_{i3} \geq 1$$

$$x_{i4} = 0 \quad \forall i \notin S_{\text{MutAid}}$$

$$z_{lt} = c_{1-lt} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + (t_{1-load} + t_a)} + c_{2-lt} \cdot \sum_{i \in S_2} x_{i1} \times \frac{\Lambda}{\tau_{2-trip} + (t_{2-load} + t_a)}$$

$$t_a = t_s \left(\frac{c}{c - \sum_i s_i (x_{i1} + x_{i2})} \right)^{\alpha}$$

$$z_{lt}^{(ex)} = y_2^{(ex)} \cdot (z_{lt} - \lambda_2)$$

$$z_{lt} - \lambda_2 \leq M y_2^{(ex)}$$

$$\lambda_2 - z_{lt} \leq M\left(1 - y_2^{(ex)}\right)$$

$$z_{svr} = c_{1-svr} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + (t_{1-load} + t_a)} + c_{2-svr} \cdot \sum_{i \in S_2} x_{i1} \times \frac{\Lambda}{\tau_{2-trip} + (t_{2-load} + t_a)} + z$$

$$z_{svr}^{(ex)} = y_3^{(ex)} \cdot (z_{svr} - \lambda_3)$$

$$z_{svr} - \lambda_3 \leq M y_3^{(ex)}$$

$$\lambda_3 - z_{svr} \leq M\left(1 - y_3^{(ex)}\right)$$

$$z_{mm} = c_{1-mm} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + \left(t_{1-load} + t_a\right)} + c_{2-mm} \cdot \sum_{i \in S_2} x_{i1} \times \frac{\Lambda}{\tau_{2-trip} + \left(t_{2-load} + t_a\right)} + z_{svr}^{(ex)}$$

$$z_{mm}^{(ex)} = y_4^{(ex)} \cdot \left(z_{mm} - \lambda_4\right)$$

$$z_{mm} - \lambda_4 \le M y_4^{(ex)}$$

$$\lambda_4 - z_{mm} \le M\left(1 - y_4^{(ex)}\right)$$

$$n_e = \left(z_{lt} - z_{lt}^{(ex)}\right) \cdot \delta_t / \Lambda$$

$$p = 1 - g^{-\lambda \cdot \delta_t}$$

$$\lambda = a^{-\left(\sum_i w_i^{(med)} x_{i2}\right)}$$

$$z \ge a_1 \cdot \sum_{i \in S_1 \cup S_2} x_{i3} + b_1$$

$$z \ge a_2 \cdot \sum_{i \in S_1 \cup S_2} x_{i3} + b_2$$

$$\cdots \cdots$$

$$z \ge a_n \cdot \sum_{i \in S_1 \cup S_2} x_{i3} + b_n$$

$$x_{ik} = \begin{cases} 1 & \text{if agent } i \text{ is assigned to task } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_2^{(ex)}, y_3^{(ex)}, y_4^{(ex)} \text{ binary}$$

6.2.2 Linearization process

In general, nonlinear programs (NLP) are harder to solve compared to linear models. Nonlinear optimization techniques are problem dependent, i.e., no one NLP solution method generally works for all the problems. Furthermore, commercial linear model solvers are much more advanced than nonlinear model solvers. In order to utilize the state-of-the-art linear solvers (e.g., CPLEX [56], GLPK [46]) to solve our optimization model, the program has to be linearized. In our NMIP, both high-order polynomial terms and polynomial terms are present. Without losing much accuracy, we want to linearize the model and solve it with much less computational effort.

The following presents some general guidelines for linearizing the commonly used nonlinear polynomial 0-1 terms (see, for details, [70, 94, 100, 101, 115, 126]).

A) For polynomial term $x_1 x_2$, where $x_1, x_2 \in \{0,1\}$. Let $x_1 x_2 = y$, and add constraints:

$$\begin{aligned} &0 \le y \le x_1 \\ &y \le x_2 \\ &y \ge x_1 + x_2 - 1 \end{aligned} \tag{6-45}$$

B) For polynomial mixed term xy, where $x \in \{0,1\}$, $0 \le y \le k$ [70, 126]. Let $xy = z$, and add constraints:

$$\begin{aligned} &0 \le z \le kx \\ &z \le y \\ &z \ge y - k(1 - x) \end{aligned} \tag{6-46}$$

C) For more general polynomial mixed term xy, where $x \in \{0,1\}$, $l \le y \le u$ [94]. Let $xy = z$, and add constraints:

$$\begin{aligned} &lx \le z \le ux \\ &z \le y - l(1 - x) \\ &z \ge y - u(1 - x) \end{aligned} \tag{6-47}$$

D) For hyperbolic (fractional) term $\dfrac{a_0 + \sum_i a_i x_i}{b_0 + \sum_i b_i x_i}$ ($b_0 + \sum_i b_i x_i \ne 0$), where $x_i \in \{0,1\}$ [94].

Let $\dfrac{1}{b_0 + \sum_i b_i x_i} = y$ (*), $l \le y \le u$. The equation (*) can be transformed to $b_0 y + \sum_i b_i x_i y = 1$. The original fractional term then becomes $a_0 y + \sum_i a_i x_i y$. We can use C) to linearize $x_i y$. Note that the variable y is regarded as a continuous variable although x_i's are discrete variables.

The piecewise linearization technique is a universal method to linearize two-dimensional functions $y = f(x)$. The idea is to divide the entire curve into several pieces and use a straight line to approximate each of the segments. Figure 6-6 shows an example polynomial function in blue and its piecewise linear approximation in red.

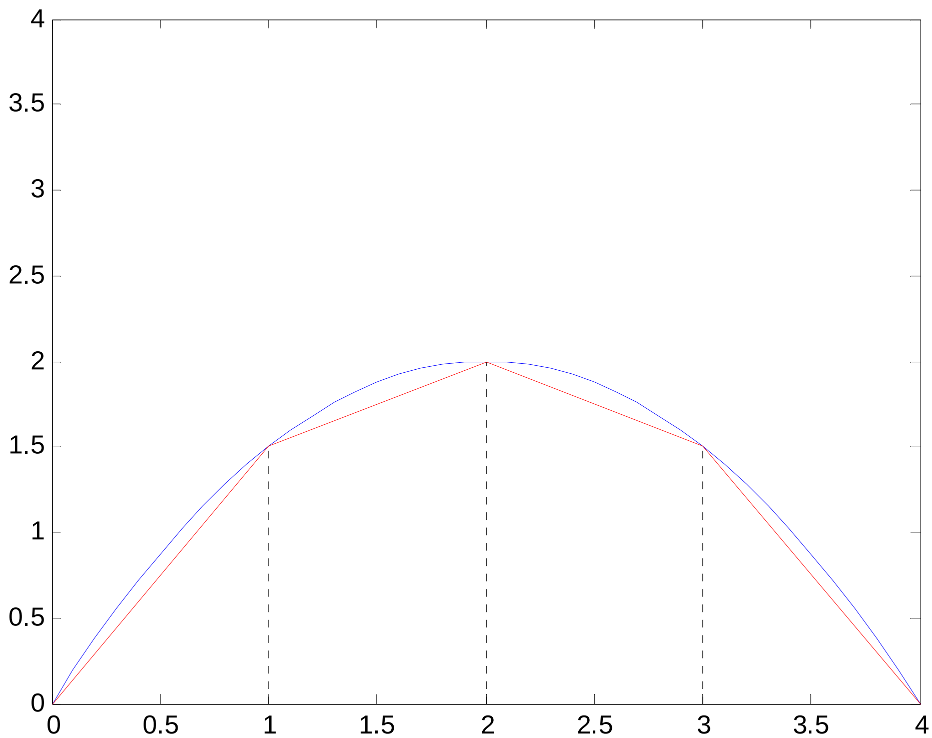

Figure 6-6. An example of piecewise linear approximation

Any two-dimensional curves $y = f(x)$ can be approximated by such sets of straight lines as $\{y = a_i x + b_i, \text{when } l_i \le x \le u_i\}$. The accuracy of the approximation increases with the number of pieces. In many cases, extra binary variables may be needed to force the functional value into the correct line segment. To implement the piecewise linear approximation, a simple routine is programmed to automatically generate the line segments and build the linearized model. Given two points (x_1, y_1) and (x_2, y_2), the through line function will be $y = ax + b$ where $a = \frac{y_1 - y_2}{x_1 - x_2}$, $b = \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2}$.

The first part of equation (6-10) is taken as an instance to illustrate the above linearization techniques. The equation is formulated as follows:

$$z_{lt}^{(1)} = c_{1-lt} \cdot \sum_{i \in S_1} x_{i1} \times \frac{\Lambda}{\tau_{1-trip} + (t_{1-load} + t_a)} \tag{6-48}$$

where,

$$t_a = t_s \left(\frac{c}{c - \sum_i s_i (x_{i1} + x_{i2})} \right)^{\alpha} \tag{6-49}$$

Let

$$y = \frac{\Lambda}{\tau_{1-trip} + (t_{1-load} + t_a)} \tag{6-50}$$

Equation (6-48) can be rewritten as:

$$z_{lt}^{(1)} = c_{1-lt} \cdot \sum_{i \in S_1} x_{i1} y \tag{6-51}$$

Because x_{i1} is a binary variable and y is a bounded continuous variable, the terms $x_{i1}y$ can be easily linearized using the general guideline C) presented in the beginning of this subsection.

The next step is to linearize the function y. Let $q_s = \sum_i s_i \cdot x_{i1} + \sum_i s_i \cdot x_{i2}$ which is a linear expression, then equation (6-49) becomes a function of q_s:

$$t_a = t_s \left(\frac{c}{c - q_s} \right)^{\alpha} \tag{6-52}$$

Substitute (6-52) into (6-50), y also becomes a nonlinear function of q_s and can be approximated by a set of piecewise linear functions of q_s. To do so, we divide the entire curve of y into J consecutive segments and approximate each segment $j \in J$ by a line $y_j = a_j q_s + b_j$. When the variable q_s falls in a segment $j \in J$ bounded by a lower bound l_j and an upper bound u_j, the function y can be approximately expressed by:

$$y \approx y_j = a_j q_s + b_j \text{ when } l_j \le q_s \le u_j \tag{6-53}$$

The following program is used to identify which segment a specific q_s value falls in:

$$\begin{aligned}
&q_s \ge l_j - M(1 - z_j) && \forall j \in J \\
&q_s \le u_j + M(1 - z_j) && \forall j \in J \\
&\sum_{j \in J} z_j = 1 && \\
&z_j \text{ binary} && \forall j \in J
\end{aligned} \tag{6-54}$$

where M is a sufficiently large constant number. The binary variable z_j is set to be 1 if and only if $q_s \in [l_j, u_j]$. So following (6-53),

$$y \approx \sum_{j \in J} z_j \cdot (a_j q_s + b_j) = \sum_{j \in J} a_j z_j q_s + b_j z_j \tag{6-55}$$

Again, because z_j is a binary variable and q_s is regarded as a positive bounded continuous variable, the terms $z_j q_s$ can be easily linearized using the general guideline B) presented in the beginning of this subsection. Up to this point, the equation (6-48) has been linearized completely.

The fourth objective function presented in (6-32) is a challenging one in terms of linearization. The entire function can be manipulated and written as follows:

$$Q_4 = \lambda_2\left(1-g^{-\lambda\Lambda}\right) + n_e \frac{1-g^{-\lambda\Lambda}}{1-g^{-\lambda\delta_t}} - n_e\Lambda/\delta_t \tag{6-56}$$

where λ and n_e are regarded as continuous variables and the others are constant parameters. If we just use the piecewise linearization method to reformulate the term $\frac{1-g^{-\lambda\Lambda}}{1-g^{-\lambda\delta_t}}$ to the linear form $a\lambda + b$, we will then obtain the multiplication of two continuous variables $n_e\lambda$ which cannot be separated easily. Some global optimization algorithms such as branch-and-bound method might have to be used to solve such kind of problems.

Instead of performing global optimization, an approximate approach is developed in this research to linearize this objective function. For the term $\frac{1-g^{-\lambda\Lambda}}{1-g^{-\lambda\delta_t}}$ (denoted as η), it has to be reformulated in such a way that it only contains a linear combination of binary variables so that the linearization method C) can be applied to separate them from n_e. Variable λ is expressed as $\lambda = a^{-\left(\sum_i w_i^{(med)} x_{i2}\right)}$. Let $p_m = \sum_i w_i^{(med)} x_{i2}$. Discretize the entire range of p_m into small intervals with a user defined length Δ and number the intervals from 0 to $q = \lfloor \max(p_m)/\Delta \rfloor$. If and only if the p_m value falls into the t^{th} interval, let $y_n = 1$, otherwise $y_n = 0$. So y_n basically servers as the interval number indicator. The above relationship is enforced by the following set of linear constraints.

$$\begin{aligned}
&\sum_i w_i^{(med)} x_{i2} = \sum_{n=0}^{q} \Delta \cdot n \cdot y_n + s \\
&\sum_{n=0}^{q} y_n = 1 \\
&0 \le s \le \Delta \\
&y_n \text{ binary}
\end{aligned} \tag{6-57}$$

Based upon the above constraints, when Δ is significantly small, p_m can be approximated as $p_m = \sum_i w_i^{(med)} x_{i2} \approx \sum_{n=0}^{q} \Delta \cdot n \cdot y_n$. Since only one y_t can be set to one, the following approximation formula can be written out:

$$\eta = \frac{1-g^{-\lambda\Lambda}}{1-g^{-\lambda\delta t}} = \frac{1-g^{-a^{-p_m}\Lambda}}{1-g^{-a^{-p_m}\delta t}} \approx \sum_{n=0}^{q} \frac{1-g^{-a^{-\Delta \cdot n}\Lambda}}{1-g^{-a^{-\Delta \cdot n}\delta_t}} \cdot y_n \qquad (6\text{-}58)$$

By this point, the complex η function with continuous variables has been reformulated as a linear combination of binary variables.

The collection of the above three methods, i.e., general guidelines A) to D), piecewise linear approximation and discretization method, forms a generic approach for the linearization of nonlinear functions. Utilizing the approach, all types of nonlinear functions discussed in this dissertation can be reformulated as sets of linear mixed integer programs within certain error ranges.

A brief summary is provided here to review the analytical optimization model formulation sections before we proceed to the computations results. In this dissertation, the simulation optimization model for disaster response management primarily deals with a nonlinear assignment problem, i.e., to assign proper tasks to the right responder agents at the right time. This optimization problem itself is NP-hard. Furthermore, because running the simulation model to evaluate the objective functions is computationally expensive, we formulate a nonlinear mixed-integer program (NMIP) which can estimate the simulation model and take the place of simulation for the optimization purpose. In order to solve the NMIP by some state-of-the-art linear solvers, the model is further linearized. The solution obtained from the analytical optimization model is expected to guide the simulation-based heuristic search into a promising region much more quickly at a lower computational cost.

6.3 COMPUTATIONAL STUDIES

6.3.1 Environment

All the following computations were performed on a personal computer (PC) with the Intel Pentium 4 CPU at 3.06 GHz and 1.00 GB of RAM memory under the Microsoft Windows XP operating system. The simulation model was implemented in Rockwell Arena 10.0 with default settings [97]; the optimization algorithm was coded in Visual Basic.NET 2003 using CPLEX 9.0 Windows API with default settings [56].

6.3.2 Pilot study

As a pilot study, a relatively small network with 20 nodes was designed to test the simulation-based decision support system for disaster management planning [125]. This is depicted in Figure 6-7. Although the network is small, both the simulation and optimization modules are fully functional.

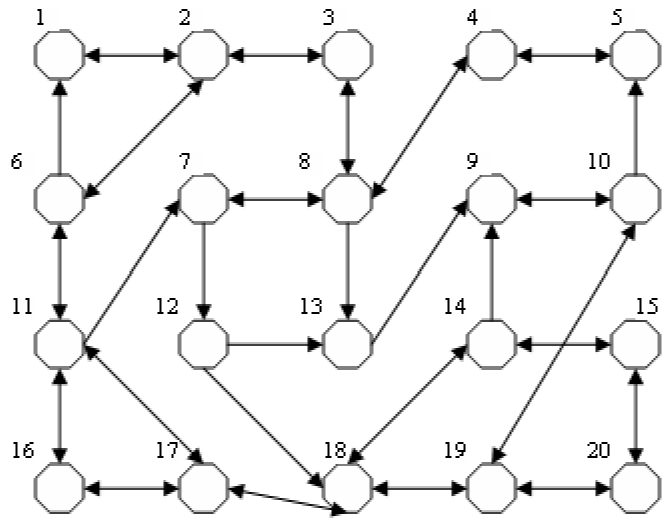

Figure 6-7. 20-node testing network

The network is completely connected which means any one node can access any other node through some finite path within the network. Some paths are one-way which are drawn by single-arrow connection lines and others are double-direction streets. Medical resources (e.g., hospitals, fire stations) are distributed on the network nodes and agent-based emergency vehicles can travel along the network from start nodes to destination nodes.

As described earlier, three main types of responder agents are included in the model: S_1 – ALS ambulances, S_2 – BLS ambulances, and S_3 – fire trucks. In this pilot study, ALS and BLS evacuate victims from the scene to medical treatment facilities (e.g., hospitals) for more definitive treatment. Fire responders, if dispatched, mainly stay at the scene, treat the patients and stabilize the situation prior to evacuation. ALS and BLS have different capabilities and efficiencies in transporting the victims as the rules state in subsection 4.2.3. The capacity parameters (as defined in subsections 6.2.1.1 to 6.2.1.3) in the NMIP model are specified as in Table 6-3. The left column regulates the evacuation capacity for an ALS vehicle per trip – it can transport one life-threatening, one severe, and one moderate patient at a time; the right column regulates the evacuation capacity for a BLS vehicle per trip – it can only transport one moderate patient at a time.

Table 6-3. Ambulance capacity parameters

ALS Capacity Parameters	Value	BLS Capacity Parameters	Value
c_{1-lt}	1	c_{2-lt}	0
c_{1-svr}	1	c_{2-svr}	0
c_{1-mm}	1	c_{2-mm}	1

Extensive simulation experiments are needed for the normal emergency responses to construct the fifth objective function of the analytical NMIP model. Based on this need, random experiments are designed and conducted to determine the relationship between normal response performance and the number of responding vehicles. In each run of the experiment, the vehicle base nodes are generated uniformly over the network and *n* vehicles (*n* ranges from 1 to 40) are uniformly chosen to respond to the normal emergency medical calls in the simulator. The normal response performance is evaluated by the degradation level which is defined as the probability that the response latency is longer than a certain amount of time (e.g., eight minutes in this context). The service performance improves when the degradation value decreases. After a significant number of replications of the experiment, a piecewise linear model can be fitted as follows.

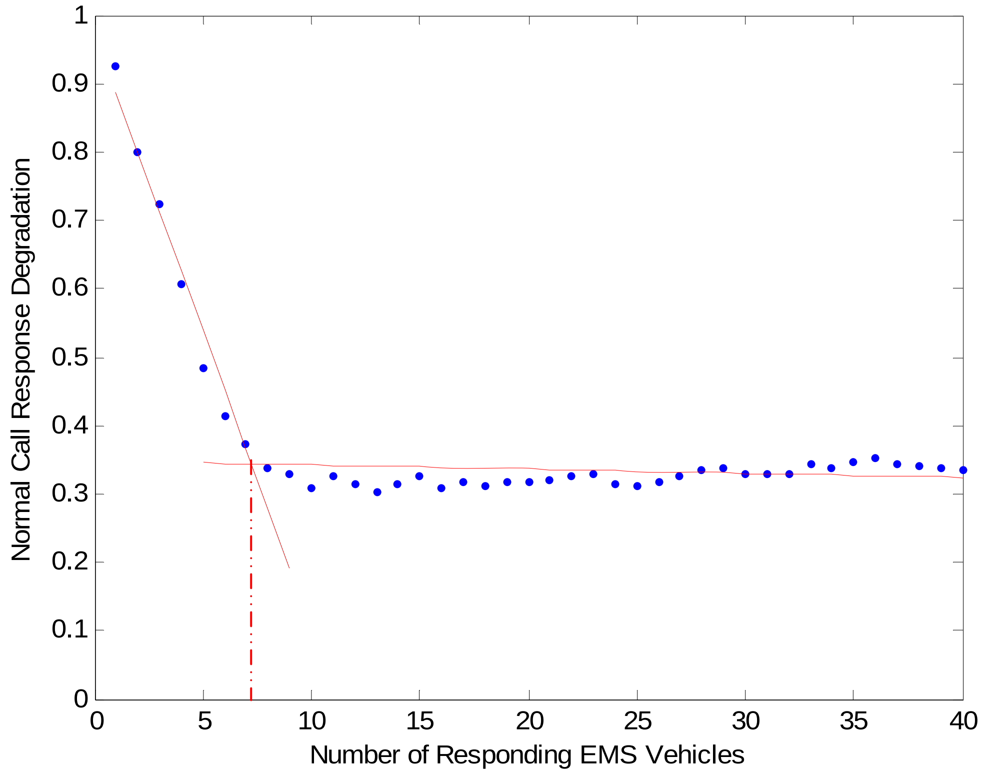

Figure 6-8. EMS normal call response degradation curve (20-node network)

$$y = \begin{cases} -0.0867x + 0.9732 & x \le 7.26 \\ -0.0006x + 0.3483 & x > 7.26 \end{cases} \tag{6-59}$$

The above results suggest that the critical response resources for EMS responding to normal calls in this problem are 8 vehicles. As can be observed from the fitted piecewise linear curve (Figure 6-8), the noise appears to be large especially in the second part of the model because the base locations of the vehicles are all randomly generated. If the base locations are fixed which is true in most city EMS systems, the noise of the model will be reduced significantly.

6.3.2.1 General test for effectiveness of analytical model

The NMIP model is now completed with all the required parameters. Next, extensive experiments are conducted to verify the effectiveness of the analytical model. The specific model settings are described as follows:

- 50 agents: 20 ALS ambulances among which 6 are from mutual aids, 20 BLS ambulances among which 8 are from mutual aids, and 10 fire trucks. The agents' base nodes, starting locations and previous response assignments are all randomly generated.
- 4 resource locations: 3 hospitals at node #1, 2 and 17 have unlimited capacities on different levels and one evacuation site at node #5 can only receive moderate victims. The resource will not be saturated during the event.
- Disaster scenarios are randomly generated within a range. The numbers of victims are uniformly generated from 100 to 200 for the levels of life-threatening, severe and moderate, respectively.

In the pilot study, 40 random experiments as designed according to the above specifications are conducted. In each of the experiments, the analytical approximate NMIP model (i.e., the linearized model) is solved initially and its solution is then evaluated by the simulation model. Figure 6-9 plots the aggregate objective values evaluated by simulation model versus by original NMIP model for all of the 40 random experiments. It basically shows that the analytical results are closely related to the simulation results because the data points marked by blue dots are around the diagonal base line. For a more extensive set of experiments, the average relative errors of original NMIP results compared to simulation results is 8.13%.

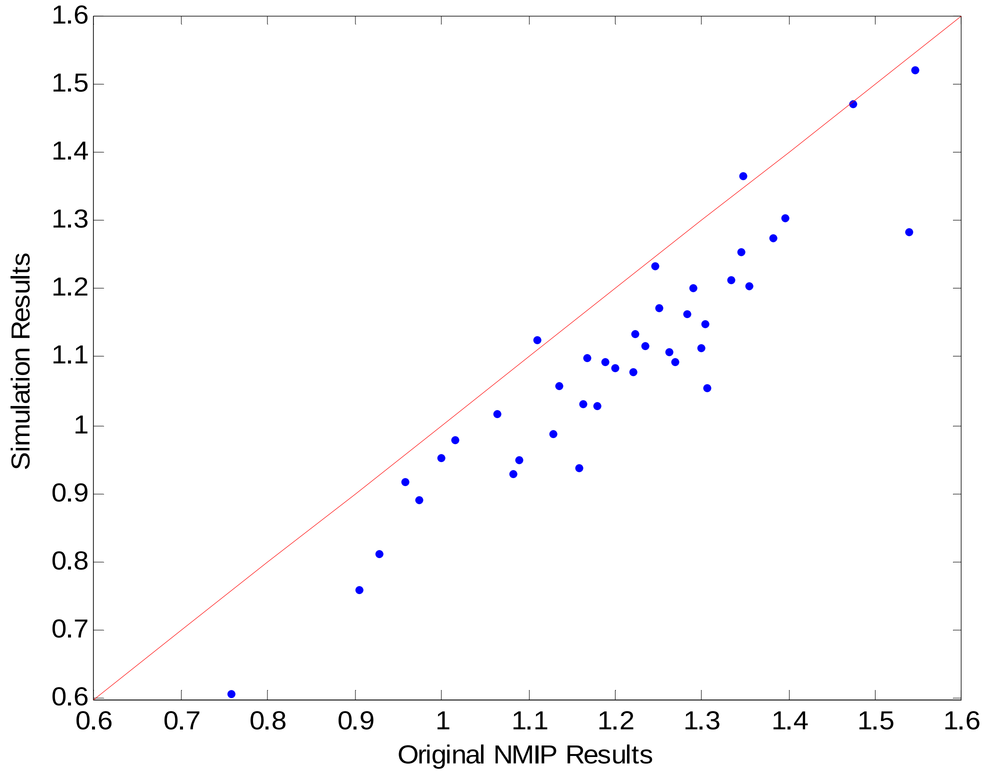

Figure 6-9. Simulation results vs. original NMIP results

Figure 6-10 plots the aggregate objective values evaluated by linearized NMIP model versus by original NMIP model for all of the 40 random experiments. It demonstrates that the linearization of the original analytical model does not affect the analytical results significantly because the data points marked by blue dots are close to the diagonal base line. For a more extensive set of experiments, the average relative errors of linearized NMIP results compared to the original NMIP results is 2.11%. Therefore, the linearization is correct and effective to some extent.

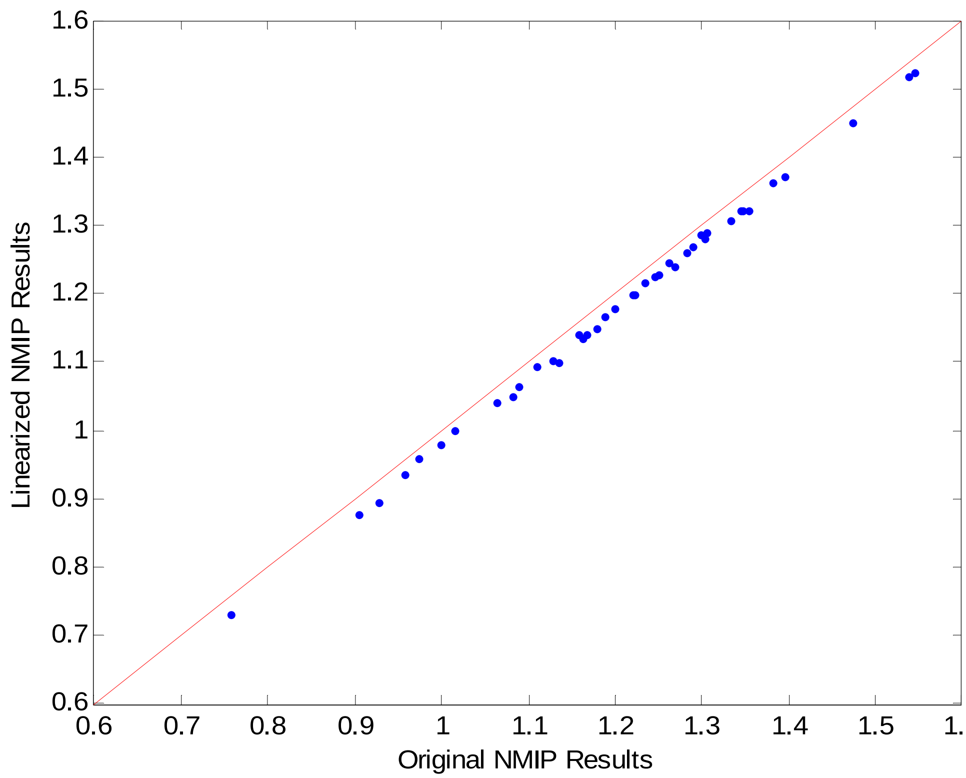

Figure 6-10. Linearized NMIP results vs. original NMIP results

Based on these results and observations, we can conclude that the analytical model (i.e., NMIP) is well formulated because it is capable of representing the trend of disaster simulation model within certain ranges. Because the analytical model can be solved in a short amount of computer time, it guides the heuristic search into a promising solution region quickly and helps make decisions in a timely manner.

6.3.2.2 Complete case study

Compared with the random simulation experiments, a thorough case study can better demonstrate the effectiveness of the evolutionary decision making procedure as a whole. A specific disaster scenario is designed as follows. 260 life-threatening, 346 severe and 223 moderate victims are involved in the event. 120 victims have already died at the beginning of the event. The disaster happens at node #4. All the other settings are the same as in the previous random experiments. The disaster decision support system is launched to generate dynamic, evolutionary response decisions every hour as the disaster event progresses until the scene is cleared. So the length of decision making period ($t_{i+1} - t_i, t_{i+2} - t_{i+1}$, …, as shown in Figure 5-1) is fixed to 1.0 hour. Also, the length of the decision evaluation horizon Λ is fixed to 4.0 hours in this study. The output of the system is essentially a dynamic responsibility chart for the responder agents as shown in Figure 6-11. This chart enables the disaster management to rationally assign responding tasks to each of the available responders and mutual aid units dynamically. Each column of the chart contains all the agents' tasks which should be assigned in that time slot. Each row of the chart shows the dynamic, time-wise response assignments for that specified agent. Note that in the chart below, the response assignments are encoded and expressed by integer numbers 1-4 which are defined in subsection 6.1.1. For example, during time 12:00 to 13:00, Agent 36-1 is dispatched to evacuate victims off the disaster scene; Agent 36-2 responds and stays at the scene to stabilize victims; Agent 37-4 and 37-6 are reserved to respond to normal emergency incidents in the area; Agent 36-7 is a mutual aid unit from outside of the area and it is not called off to respond to the disaster.

Agent \ Time	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
36-1	1	2	1	2	3	2	1	1	3	1
36-2	2	2	1	1	3	2	1	2	3	1
36-3	1	2	1	2	3	2	1	2	3	1
36-4	1	2	1	1	3	2	1	2	3	1
36-5	1	2	1	2	3	2	1	2	1	1
36-6	1	2	1	1	3	2	1	2	2	1
36-7	4	3	1	2	3	2	1	1	2	1
37-1	4	3	1	1	4	2	1	1	2	1
37-2	1	3	1	2	4	2	4	1	2	1
37-3	1	3	1	1	3	2	4	1	2	1
37-4	3	3	1	2	3	2	4	3	2	1
37-5	2	3	1	1	4	2	4	3	1	1
37-6	3	3	1	2	3	2	4	3	1	1
37-7	2	3	1	1	3	2	1	1	1	2
37-8	1	1	1	2	3	1	1	1	1	2
37-9	1	1	1	1	3	1	1	1	1	2
38-1	1	1	1	2	3	1	1	3	1	2
38-2	1	1	1	1	3	1	1	1	2	2
38-3	1	1	1	2	3	1	1	1	2	2

Figure 6-11. Sample dynamic responsibility chart for disaster responders

The dynamic response solutions are compared against a fixed solution provided by experts and/or protocols. In the current practice, to respond to a mass-casualty major disaster, most of the available responders are dispatched to the disaster scene immediately with only several reserved for other purposes, e.g., responding to other normal emergency incidents [110]. This expert opinion is used as the fixed solution for this study. Figure 6-12 compares the aggregate multi-objective values between the dynamic solutions obtained by the evolutionary decision procedure and the fixed expert decisions in the whole time series. Since the time between decisions are fixed to 1.0 hour in this study, the CPU time for computing every new decision here is less than one hour. Note that Figure 6-12 indicates that for this minimization problem, the dynamic response solutions always obtain better overall performance.

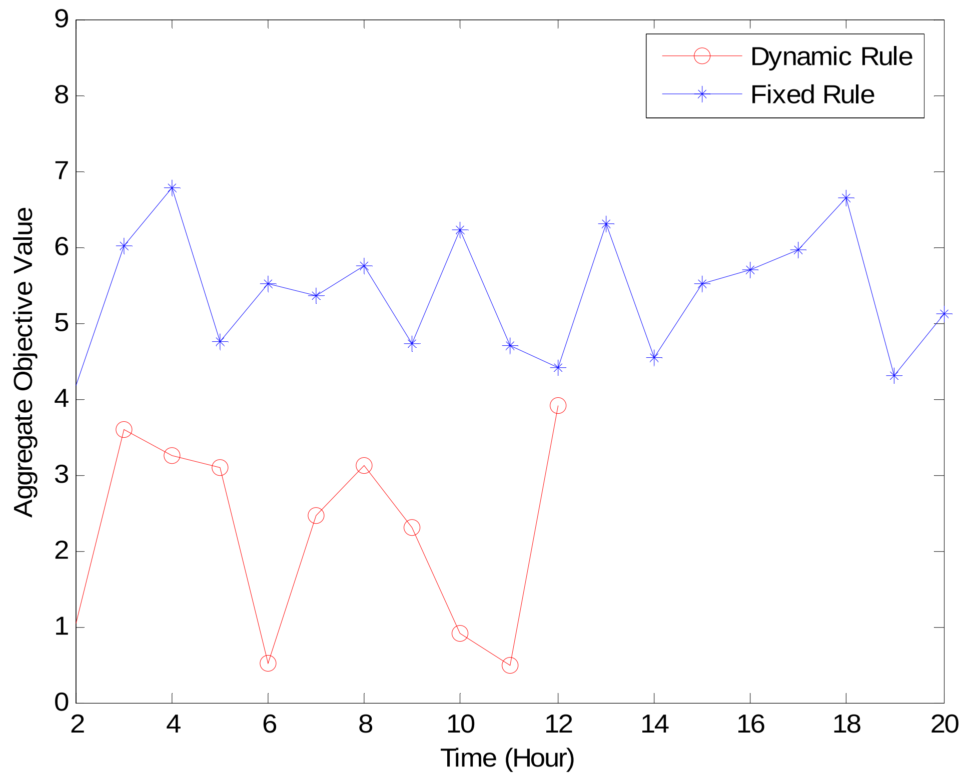

Figure 6-12. Comparison of aggregate objective values

It is hard to interpret the aggregate objective values because they do not have physical meanings. To better understand the dynamic solutions and their effectiveness, the key individual objectives are extracted in the following series of figures. Figure 6-13 shows the evacuation situation of life-threatening (LT) victims. With the dynamic solutions, LT patients can be cleared by the 11th hour while it takes almost 19 hours using the fixed rule solutions.

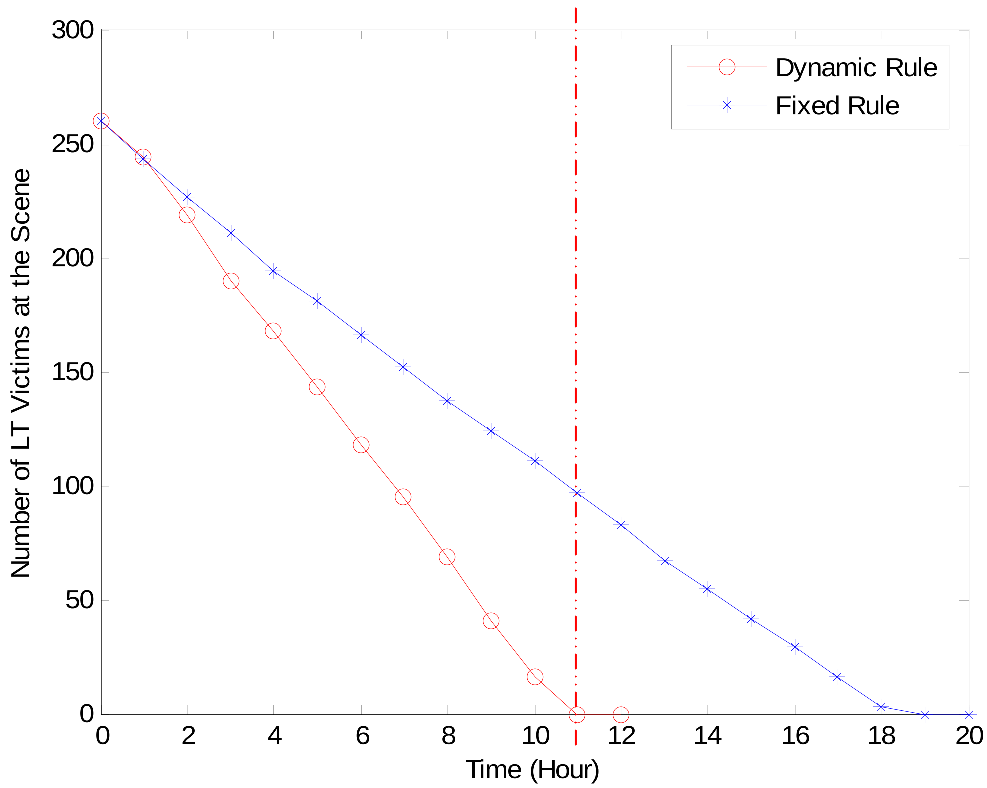

Figure 6-13. Comparison of on-scene life-threatening victim evacuation

Figure 6-14 shows the evacuation situation of severe victims. The dynamic response decisions can help evacuate severe patients eight hours faster.

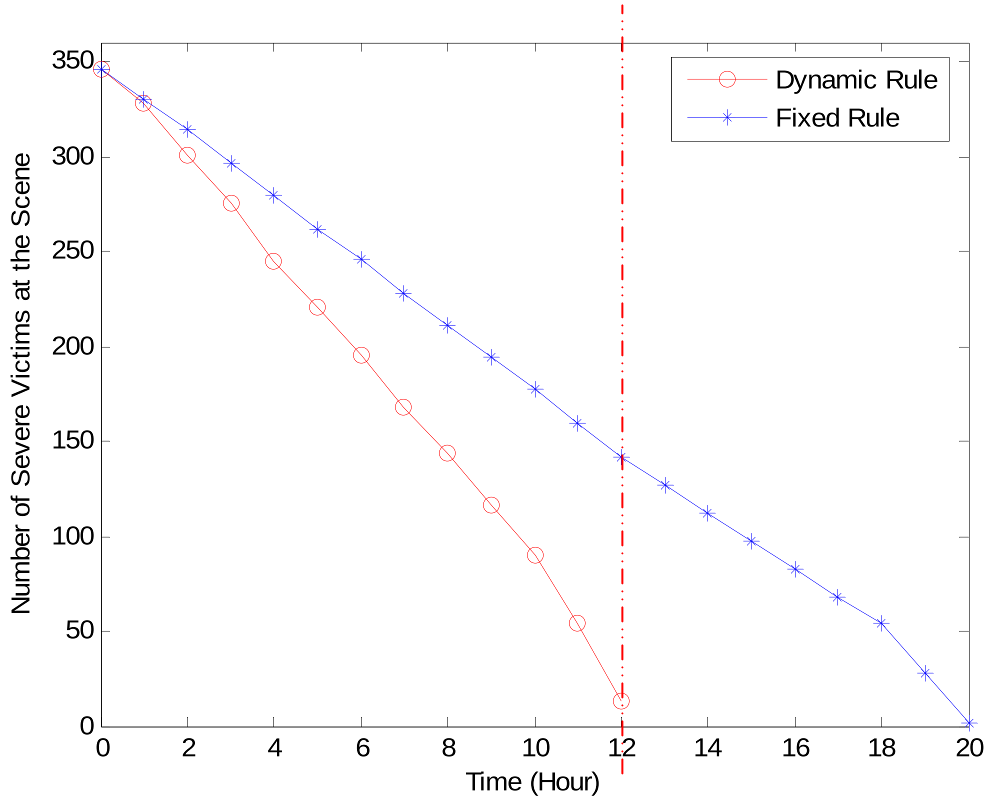

Figure 6-14. Comparison of on-scene severe victim evacuation

The moderate victims are not severely injured. They have low possibility of mortality so a lower weight is imposed on their evacuation in order to assign more resources to the LT and severe patients' treatment and evacuation. Figure 6-15 shows the evacuation situation of moderate victims. There is no significant performance improvement with the dynamic response decisions compared to the fixed rule solutions over time.

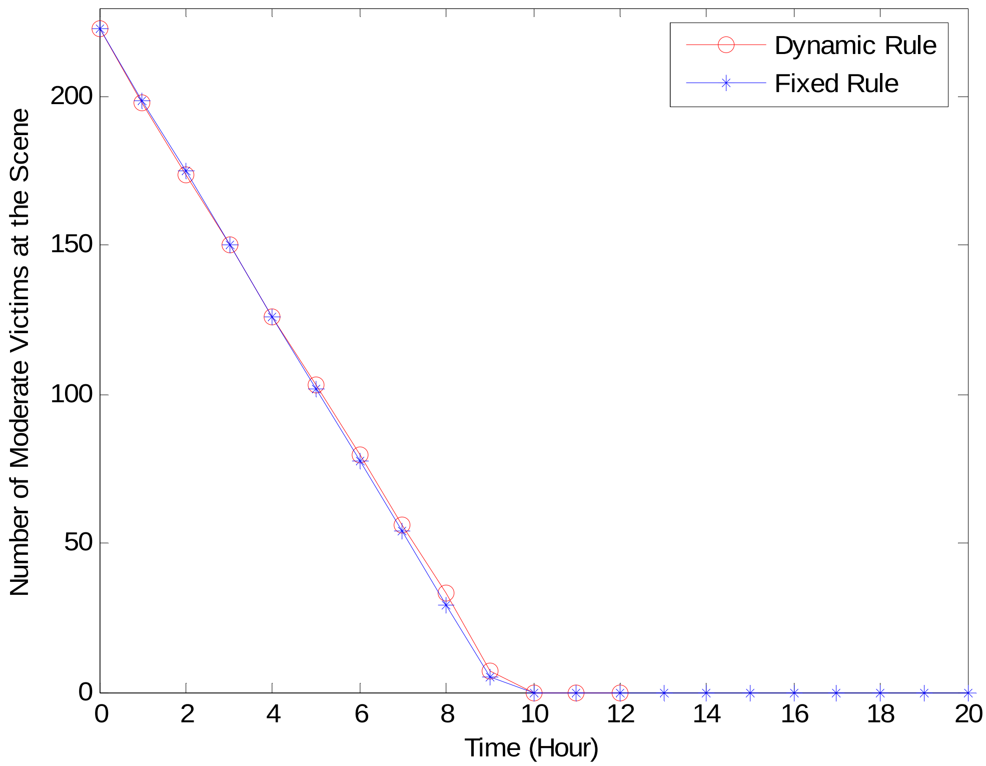

Figure 6-15. Comparison of on-scene moderate victim evacuation

Besides evacuation time, the number of fatalities at the scene is another important measure of the response effectiveness. Figure 6-16 compares the fatalities between dynamic solutions and fixed solutions. Although the death rate under the dynamic solution is initially higher, the life-threatening victims are evacuated much faster so five more lives are projected to be saved eventually compared to the fixed rule solution. By investigating the response solutions, we can find the intuition behind this. Although the fire responders can help treat/stabilize the severe victims and retard the deterioration, their appearance at the scene causes much congestion and delays the EMS's evacuation activity. Thus, the dynamic decision system dispatches the space-consuming fire trucks more conservatively so that the victims can be evacuated faster although the on-scene severe patients' decay is initially at a higher rate.

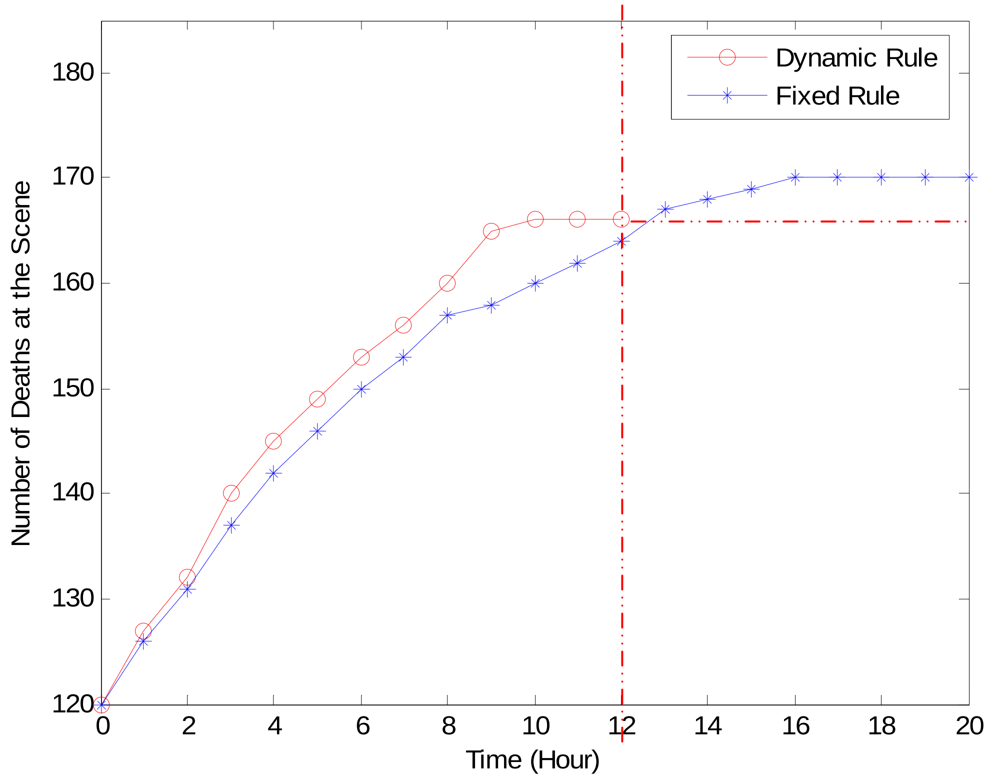

Figure 6-16. Comparison of on-scene fatalities

Last but not least, the normal call response degradation is compared in Figure 6-17. The dynamic solutions may obtain lower normal call response degradation levels compared to the fixed expert solution because the normal emergency coverage typically may be overlooked by human decision makers during a large-scale disaster event.

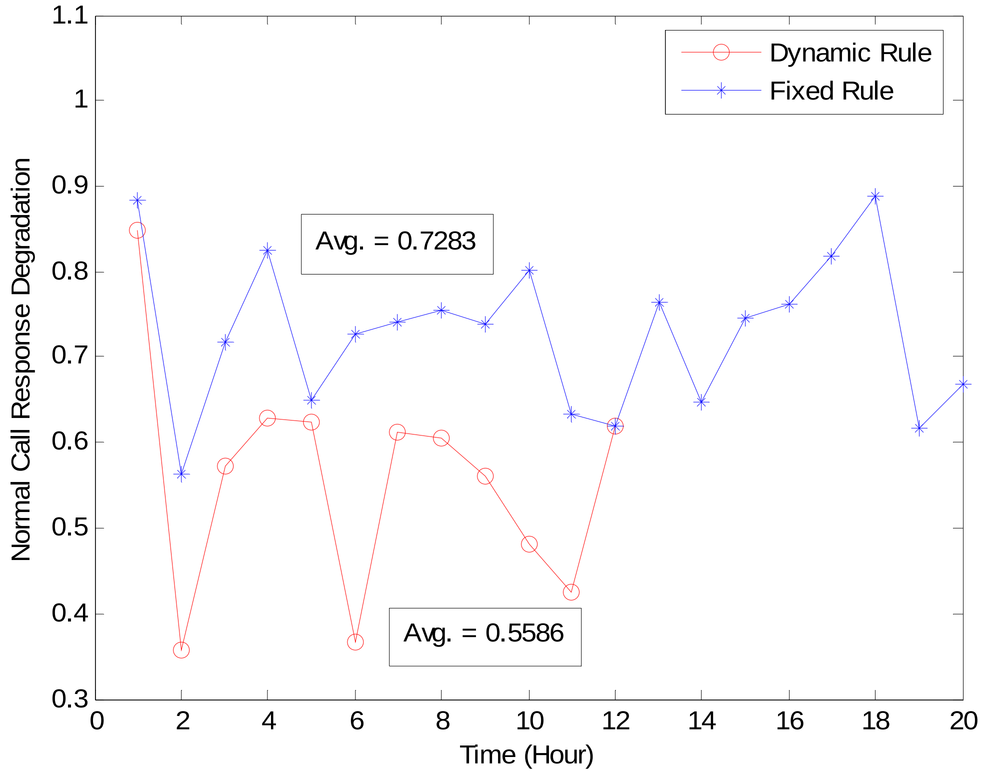

Figure 6-17. Comparison of normal call response degradation

As a short summary, the dynamic MIP solution shortens the evacuation time for life-threatening victims by 42.1%, speeds up the evacuation of severe victims by 38.1%, and reduces the fatalities occurring at the scene by 8%. Why do the dynamic decisions obtained from our decision support system outperform the fixed rule/expert decisions which have been used in disaster the response practice for a long time? When looking into the problem more closely, we can find an important insight for the large-scale disaster management. With the dynamic decision system, we have optimized the outcomes by reducing scene congestion. A more effective responder structure is achieved with 60% more ALS, 98% fewer BLS and 49% fewer fire responders. ALS responders can treat and evacuate the LT and severe patients better so they are dispatched in greater volume; fire trucks can congest the scene more easily and affect other responders' evacuation efficiency due to the large vehicle size so they are dispatcher in smaller volume. Managing the scene congestion and team efficiency properly becomes imperative in the large-scale disaster events.

6.3.3 Large-scale extended study

The above pilot study has shown the effectiveness of our disaster decision support system. In this subsection, a more sophisticated and realistic network model is presented as well as some significant findings from the computational results. This extended study aims at addressing the disaster response decision issues for the Pittsburgh downtown area so the simulation-based decision support system implemented here is called the Pittsburgh framework. Its network contains 200 nodes which are the critical intersections extracted from the Pittsburgh GIS database. The nodes are connected by main roads, highways and bridges. All of the network attributes refer to the actual GIS database including node positions, connectivity, speed limits, medical response vehicles and hospitals. Figure 6-18 shows the full 200-node network model constructed in Arena 10.0.

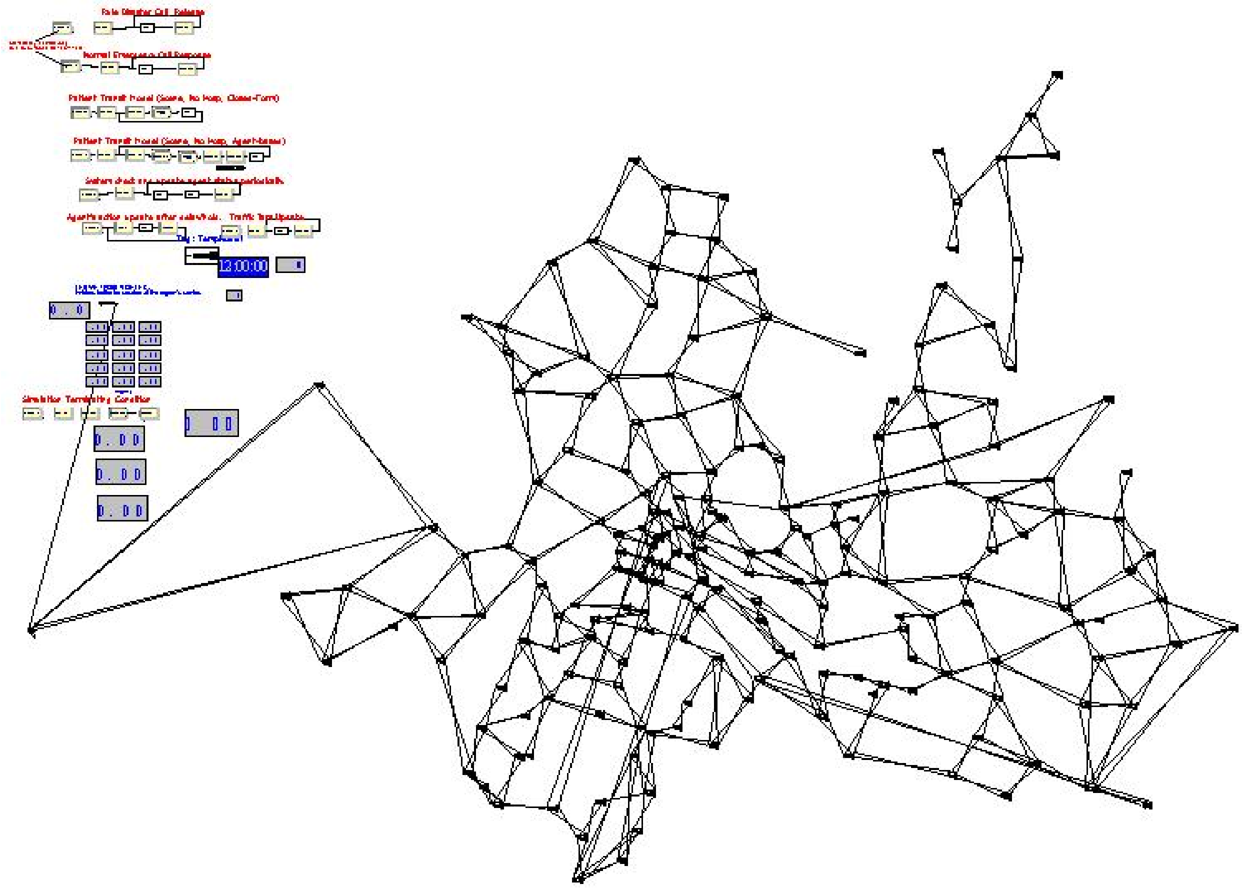

Figure 6-18. Pittsburgh 200-node full model built in Arena

First of all, we need to collect more realistic data about the region's medical call volume. If the population has been relatively stable, the call volume data should not change too much even over an extended period. Consequently, part of the data used with RURALSIM [105] are recovered and used for this study. RURALSIM's call volume data were compiled from the actual data of the Pittsburgh Emergency Medical Services. They include six categories of calls: cardiac, trauma, non-cardiac/non-trauma, motor vehicle accident, dry run and standby, and also consider some other adjustment factors such as demand variation and weekday/weekend multipliers. For the sake of simplification and testing, we averaged the call data from all categories and obtained a unified call distribution regardless of the type of patients. The averaged data are presented in Table 6-4. The normal emergency calls are randomly generated over the network based on this data. Note that a time of day multiplier could be introduced into the average data to differentiate the call volume over a day. The peak demand (e.g., for rush hours) would be about 1.5~2.5 times of the average demand.

Table 6-4. Average Pittsburgh call volume

	Call Rate	Probability of Life-Threatening	Probability of Severe	Probability of Moderate/Minor
Per Minute	0.1099	0.1519	0.0594	0.7887
Per Hour	6.5963			

As was the case for the previous model, ALS ambulances, BLS ambulances and fire trucks are the three types of response agents running in the network. They have 20, 8, and 10 units, respectively. To implement the analytical optimization model described in subsection 6.2.1.5, a piece-wise linear model is fitted by the data from extensive normal call response simulation experiments. The model identifies the relationship between the response degradation level (i.e., probability that response latency is longer than eight minutes) and the quantity of responding units. The experiment results and the fitted piece-wise linear model are presented in Figure 6-19 and Equation (6-60).

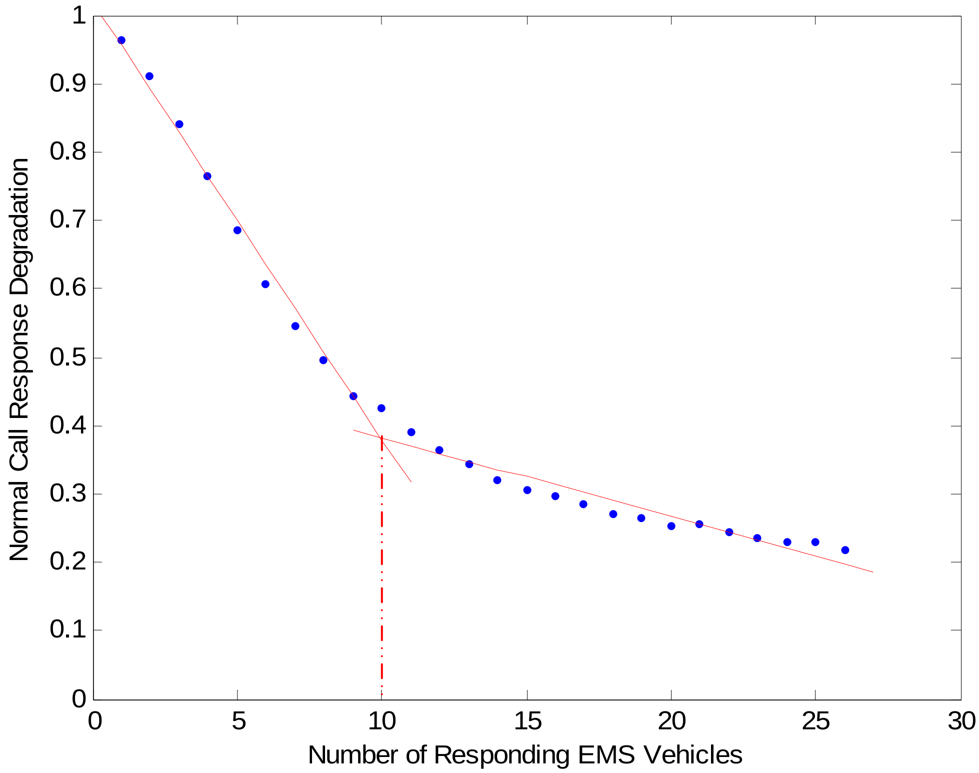

Figure 6-19. EMS normal response degradation curve (200-node network)

$$y = \begin{cases} -0.0641x + 1.0205 & x \leq 9.95 \\ -0.0115x + 0.4975 & x > 9.95 \end{cases} \tag{6-60}$$

Where y is the normal call response degradation level and x is the number of responding EMS vehicles. Therefore, the critical EMS resources for responding to normal calls in Pittsburgh are 10 vehicles.

6.3.3.1 Case study under the Pittsburgh framework

In this case study, we want to simulate an imaginary mass-casualty disaster event in the heart of the Pittsburgh downtown area at the similar scale of the 2004 Madrid bomb disaster. On March 11, 2004, serious terrorist bomb explosions occurred in Madrid, Spain with 177 people dead instantly and more than 2000 injured [91]. Based on the reported casualty data from the Madrid bomb explosions event, a specific victim distribution is designed for our case. 180 life-threatening, 373 severe and 957 moderate patients are involved; 177 victims are dead on arrival. The developed disaster decision support system is used to generate dynamic, good-quality response decisions (i.e., dynamic MIP solutions) every hour as the disaster event unfolds until all the victims are cleared from the scene. A fixed expert response solution (derived from [110]) provides a benchmark to compare with the dynamic response decisions. There are totally 20 ALS, 8 BLS, and 10 fire engines available in the system including the local resources and mutual aid units. The fixed expert decision dispatches most available emergency vehicles to respond to the disaster, leaving only three ALS ambulances to cover other incidents. Since the decision making period is fixed at 1.0 hour, the CPU time for the developed decision support system to generate a new decision is less one hour.

Figure 6-20 shows the evacuation situation of life-threatening (LT) victims. Because the fire engines can easily congest the narrow downtown streets and the fire responders are not sufficiently trained to stabilize the LT patients at the scene in order to reduce the mortality rate, the dynamic MIP solutions suggest that the fire engines should not be dispatched as first responders. This situational decision helps to reduce the disaster scene congestion and consequently enhance other responders' efficiency. Therefore, the life-threatening victims are evacuated from the scene one hour faster with the dynamic MIP solutions compared with the fixed expert decision as illustrated in the figure.

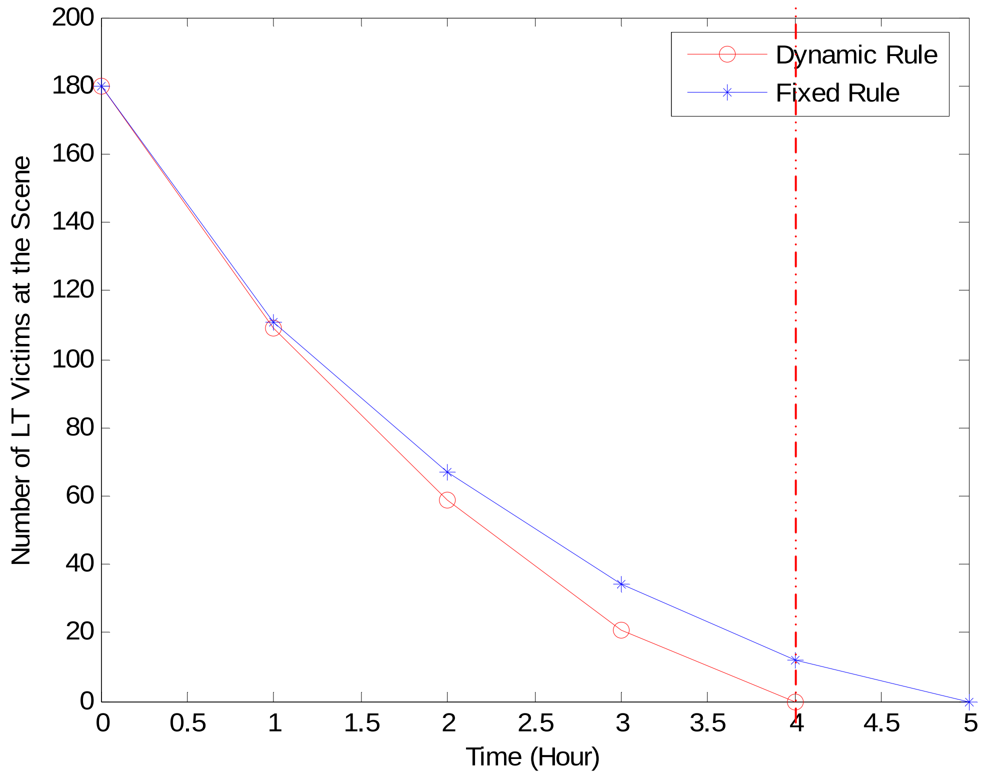

Figure 6-20. Comparison of on-scene life-threatening victim evacuation (Pittsburgh case)

Figure 6-21 shows the evacuation of severe victims. Because the scene congestion is alleviated by dispatching fewer emergency vehicles, the dynamic MIP-based response decisions can help clear the severely injured patients from the scene two hours faster.

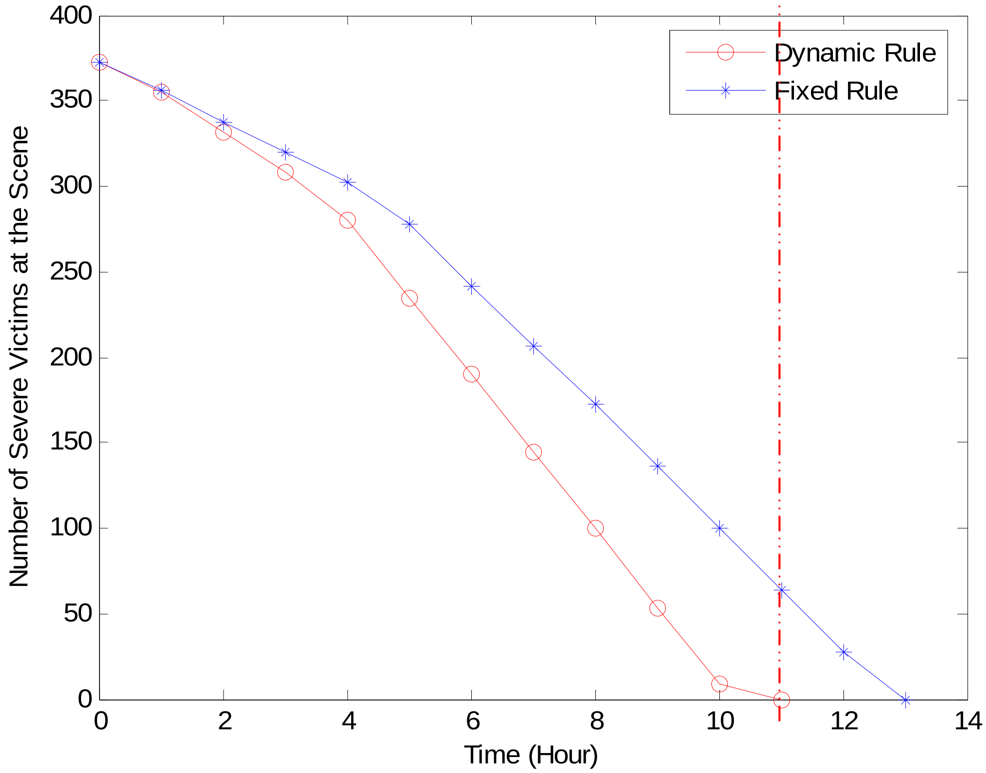

Figure 6-21. Comparison of on-scene severe victim evacuation (Pittsburgh case)

Dispatching more or fewer responders represents a tradeoff in disaster response decision making. More responders would increase the scene congestion and consequently affect the responders' overall efficiency; on the other hand, fewer responders might result in an insufficient number of responders (stabilization and/or evacuation), and consequently increase the mortality rate. The comparison of the number of deaths between the dynamic MIP solutions and the fixed expert decision as in Figure 6-22 demonstrates such a tradeoff. Under the dynamic solution, the death rate is initially higher because fewer responders are dispatched, but eventually more lives are saved because the severely injured people are evacuated faster as shown previously.

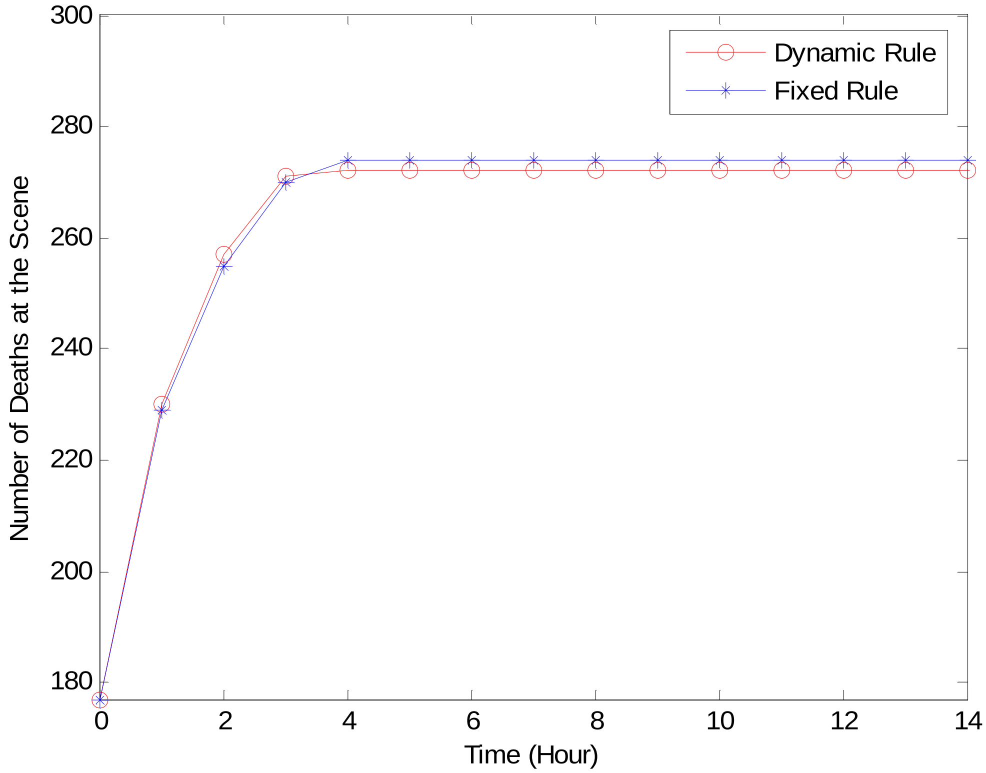

Figure 6-22. Comparison of on-scene fatalities (Pittsburgh case)

Assigning fewer unnecessary responders to the disaster scene and dispatching them to respond to normal incidents instead not only reduces the scene congestion, but also secures better coverage of the rest of the area. Figure 6-23 compares the normal call response degradation under the two sets of decisions. The dynamic solutions achieve the average response degradation of 0.34 while the fixed expert decision performs at the average level of 0.77. This case study further demonstrates the advantages of the developed dynamic decision support system by balancing various factors involved in the disaster response so as to optimize the overall performance of the system.

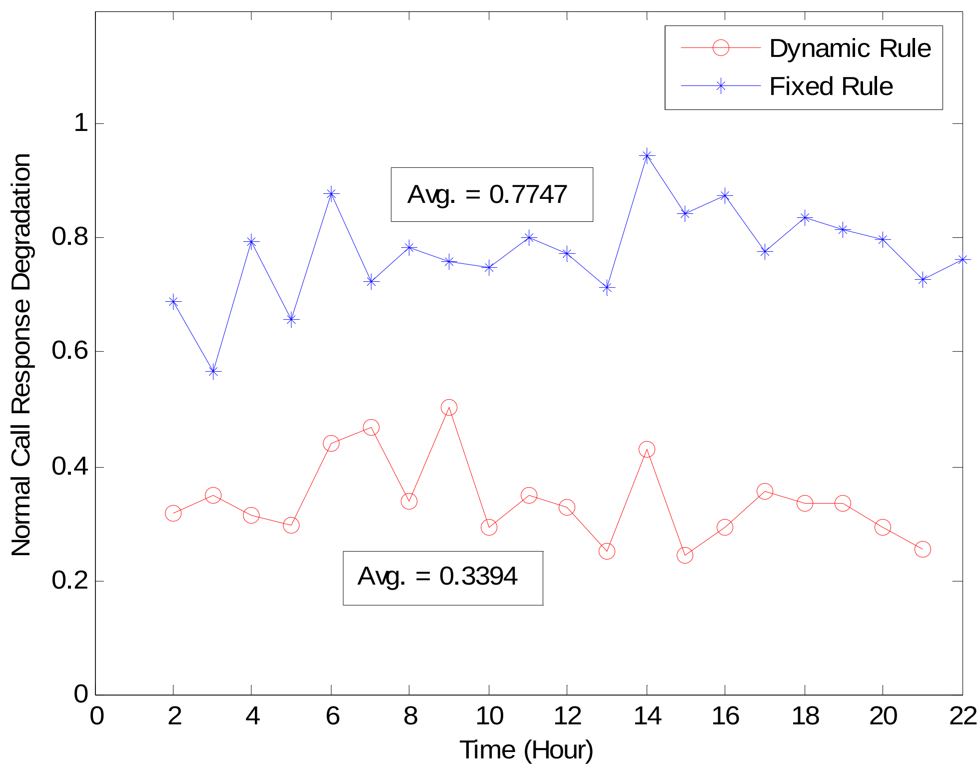

Figure 6-23. Comparison of normal call response degradation (Pittsburgh case)

6.3.3.2 Validation of the hypothesis about decision time parameters

In subsection 5.3.2, a hypothetical statement was stated as: if the decision evaluation horizon is longer than the next decision making period, the evolutionary decision making procedure is expected to provide better solutions than that with shorter evaluation horizons. In the mathematical form, it is preferred that $\Lambda_i > t_{i+2} - t_{i+1}$ in decision stage *i* (refer to Figure 5-1).

In this study, we fix the length of decision making period $\left(t_{i+2} - t_{i+1}\right)$ at 1.0 hour and vary the length of decision evaluation horizon Λ_i to see how Λ_i impacts the outcomes (aggregate objective values). A set of random disaster scenarios are constructed for running the tests. The same set of experiments are conducted for different Λ_i values so that the results are comparable. For each Λ_i, we use the results (aggregate objective values) obtained when Λ = 1.0 hour (i.e., when $\Lambda_i = t_{i+2} - t_{i+1}$) as the performance evaluator to calculate the relative improvement ratios for the various scenarios in the experiment set. The ratio (for the minimization problem) is calculated by the following formula:

$$\text{Relative Improvement Ratio of Current Result} = \frac{(\text{Result for } \Lambda = 1.0) - (\text{Current Result})}{|\text{Result for } \Lambda = 1.0|} \qquad (6\text{-}61)$$

Figure 6-24 shows the average improvement ratios for different Λ values. The average improvement ratio increases with the length of decision evaluation horizon until a peak occurs at $\Lambda = 3.5$ hours and then it decreases. Those results validate the previous hypothesis about the decision time parameters in the evolutionary decision procedure. When decisions are optimized for a time horizon longer (but not too long) than the time between decisions, the procedure can produce better performance than more near-sighted schemes. This interesting finding here introduces a potential research topic of the optimization of decision time intervals, i.e., what are the optimal intervals for the time between decisions and decision evaluation horizon, respectively?

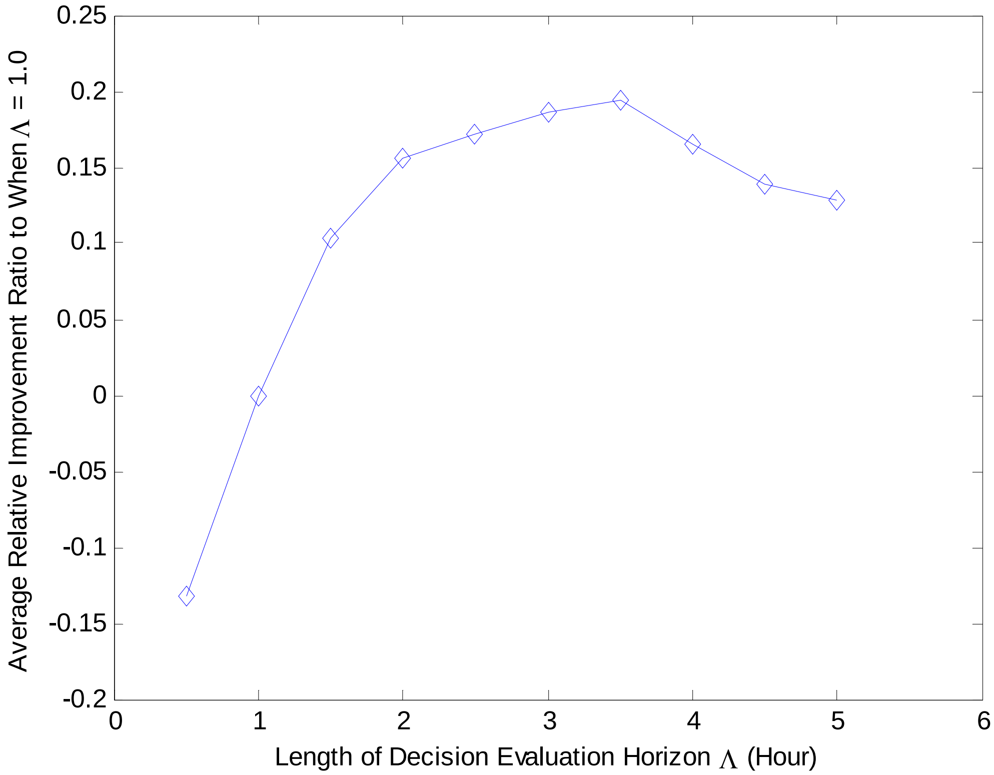

Figure 6-24. Average relative improvement ratios for different Λ's

6.3.3.3 Comparison of traditional local search and AGRS

Chapter 5.0 proposed two versions for the evolutionary decision making procedure. One contains a traditional local search algorithm using an approximate analytical solution as the initial start point (see section 5.2); the other develops a unique heuristic search algorithm called analytically guided randomized search (AGRS, see section 5.5). This subsection compares their performance through extensive random experiments.

In the AGRS algorithm, the analytical optimization model's objective functions are perturbed by their estimation errors against the simulation, i.e., *e* terms in equation (5-1). The error terms are drawn from certain distributions in each of the search iterations. So before running this heuristic online, extensive simulation runs with random input parameters should be performed offline and the experimental data are collected for fitting the appropriate estimation error distributions. The resultant error distributions are listed in Table 6-5.

Table 6-5. Analytical model estimation error distributions

Objective Function	**Estimation Error Distribution**	**Distribution Parameters**
#1: Scene evacuation number of life-threatening victims	Normal distribution	μ : 0.924, σ : 0.0782
#2: Scene evacuation number of severe victims	Empirical distribution	(P_k , V_k): cumulative probability and associated value pairs.
#3: Scene evacuation number of moderate victims	Empirical distribution	(P_k , V_k): cumulative probability and associated value pairs.
#4: Scene fatalities	Empirical distribution	(P_k , V_k): cumulative probability and associated value pairs.
#5: EMS normal incident response degradation	Empirical distribution	(P_k , V_k): cumulative probability and associated value pairs.

It is noted that objective functions 2 through 5 do not fit any theoretical statistical distributions so discrete empirical distributions are applied. The empirical distributions are modeled in discrete pairs of cumulative probability and its associated value.

Then, a set of experimental disaster scenarios are constructed. The number of victims is uniformly generated from 0 to 300 for the levels of life-threatening, severe and moderate, respectively. This set of experiments is tested on the AGRS algorithm first. Each experiment contains eleven search iterations (one original-objective plus ten perturbed-objective iterations) and is completed within one hour of CPU time. With the same amount of CPU time, the identical set of experiments is then tested using the traditional local search algorithm. The relative improvement ratios of the AGRS-based procedure compared to the traditional local search based procedure for each of the experiments are illustrated by the bar plot in Figure 6-25.

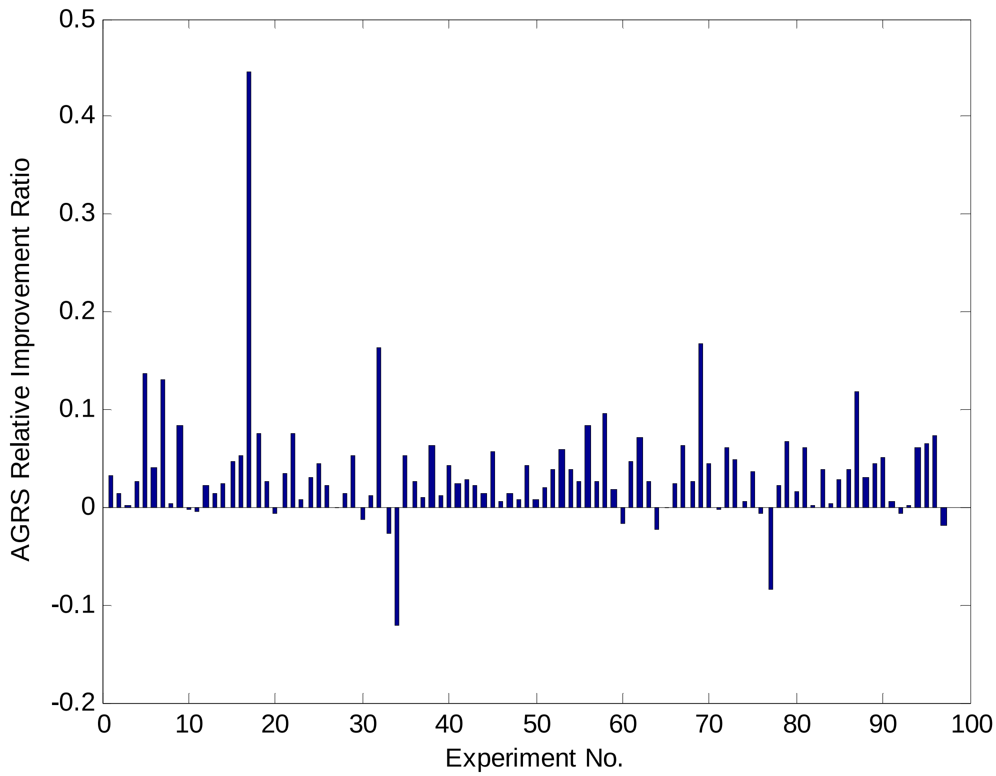

Figure 6-25. AGRS relative improvement ratios

The above bar plot includes the results from 100 random-scenario experiments in order to compare the aggregate objective values obtained by AGRS-based and local search based procedures. The plotted AGRS Relative Improvement Ratio for each pair of the experiments is calculated as follows (for the minimization problem):

$$\text{AGRS Relative Improvement Ratio} = \frac{\text{Local Search Result - AGRS Result}}{\left|\text{Local Search Result}\right|} \tag{6-62}$$

From the figure, it is obvious that the AGRS outperforms traditional local search because most improvement ratios are positive. The average AGRS relative improvement ratio for this study is

3.59%. At a 95% confidence level, the confidence interval for the AGRS relative improvement ratio is [0.024, 0.047]. Thus the improvement ratio is statistically positive. This result partly attributes to the fact that, on the average, the AGRS procedure can shun some local optima and keep improving the solutions for 4.67 iterations while the traditional local search is more easily trapped at certain local optima because it stops improving after only 1.4 iterations. This observation further justifies the importance of solution diversification for enhancing heuristics performance. More specifically, AGRS diversifies the search region by perturbing the objective functions in each iteration, while the traditional local search intensifies the initial solution by searching around its neighborhood regions. This diversification scheme can help the heuristic search to escape from local optima and seek for the global optimum.

6.4 SUMMARY

This chapter implements the evolutionary decision making procedure developed in the previous chapter to specifically solve disaster response decision problems. The construction of the approximate analytical model is the crux of the algorithm. The model formulation and linearization processes have been fully described as a comprehensive instance.

Computational results from random experiments and a complete case study demonstrate the effectiveness and efficiency of the heuristic procedure from various aspects. The embedded traditional local search and analytically guided randomized search (AGRS) are also compared through extensive random experiments. AGRS is shown to outperform the traditional local search procedure. In addition, a hypothesis about the length of decision evaluation horizon Λ (see subsection 5.3.2) is validated by experiments to some extent.

The most important achievement in this chapter is to demonstrate that our evolutionary decision making procedure is a feasible approach to optimizing large-scale stochastic operational systems in a timely manner, even in real-time.

7.0 CONCLUSIONS AND FUTURE RESEARCH

This chapter summarizes the major development, implementations and related findings in this dissertation, corresponding to the research problems and proposed contributions laid out in the first chapter. Also presented are constructive future research directions. These ideas and concepts, which could be applied to extend the simulation-based decision support system, originated in conducting this project.

7.1 SUMMARY AND CONCLUSIONS

Simulation is an effective tool for modeling complex systems such as the disaster response system without imposing overly simplified assumptions. It incorporates the stochastic nature of such systems so as to make the models more robust and convincible in practice. Discrete event simulation (DES) is a widely adopted method to model system operations in a discrete time manner. Agent-based simulation (ABS) is another rising method to model entities as autonomous decision-making individuals. ABS focuses more on modeling subtle agent rules. It lets individual agents' rule-based actions and interactions determine the system behaviors as a whole in a semi-continuous time fashion. This concept is more consistent with the real-world operations. However, ABS is not as computationally efficient as DES because the agents need to check their status, environment and predefined rules and update system states more frequently. Smaller update time intervals increase computational intensity. This dissertation combines the two simulation methods in order to utilize the advantages of both and lets them complement each other in an integrated way. With the unique agent-based discrete event modeling data structure, the simulated entities operate in an efficient discrete-event framework while their operational decisions are autonomously determined by themselves based on the perception of the environment and the compilation of predefined rules. Such logical rules are maintained outside of the simulator so that they can be modified more easily and flexibly without affecting the simulator much. Furthermore, other components/modules such as geographic information systems (GIS) and databases can be easily integrated in such a framework to impact the simulated environment and affect the execution of agent rules as needed by reality. In our study, a *Dynamic Discrete Disaster Decision Simulation System* (D^4S^2) is developed using this hybrid modeling method. It consists of an agent-based discrete event simulator, a GIS, relational databases, optimization and decision modules and user interfaces. Such an integrated, dynamic simulation system is shown to outperform a traditional stand-alone, hard-coded disaster simulation model with respect to both computational time and functionalities. This demonstrates the efficiency and effectiveness of the marriage of agent-based modeling and discrete event simulation.

Any simulation model has to be validated before it is applied. Normally the simulation validation is done by comparing simulation results with actual system statistics under certain settings. This could be extremely hard for low-probability and high-impact events such as disasters because historical data are generally incomplete or unavailable, and experiments cannot be performed on the actual systems. This dissertation describes several validation strategies for rare-event simulations. Some experimental results of D^4S^2 have been presented to subject matter experts (SMEs) for visual validation. Those results are consistent with their professional experiences so the system has face validity. Among all of the validation approaches, the theory based validation method is unique. A fictitious square flat city network is artificially created and different scenarios are tested on it. Theoretical formulations and predictions exist for such symmetric network problems. We can compare the simulation results against those theoretically correct results to assess the simulation validity. All of the validation experiments testify to the correctness of the model.

Besides system modeling and validation, the development of an efficient simulation optimization approach called the *Evolutionary Real-time Decision Making Procedure* is another major contribution of this research. This procedure aims at providing a decision support tool for management to make robust simulation-based decisions in real-time. Complex systems normally evolve over a considerable amount of time during which new situations may arise. Simulating an entire complex event from the beginning towards the end without inputting new information is neither systematically correct nor computationally efficient. The evolutionary procedure decomposes the entire process horizon into smaller time intervals and then simulates each interval in sequence, allowing system updates in the small intervals. This scheme enables a simulation system to import stochastic situational changes during the simulated events such that it incorporates another layer of reality. The shortened simulation runs also make simulation-based real-time optimization possible. As the simulated system changes, the optimal decisions to the system should also be modified accordingly. Since the entire event horizon has been divided into smaller intervals, we are able to optimize for every individual interval and obtain dynamic solutions over time. In real-time management, traditional simulation optimization heuristics do not function well because they require more simulation evaluations (runs) than allowed within a limited time frame. This research utilizes an analytical model to estimate and replace the expensive simulator in performing optimization and providing a good initial solution. The initial solution can drive the simulation-based search into a promising solution region quickly, followed by a simple, traditional local search to refine the initial solution. An enhanced algorithm called *Analytically Guided Randomized Search* (AGRS) is also designed to perturb the analytical model and shoot for multiple promising solution regions in order to gain global optimality.

The disaster response problem is used to realize the evolutionary real-time decision making procedure and verify the decision support system's effectiveness and efficiency. The test results show that the analytical model estimates the simulation well so the system is capable of producing good management decisions dynamically within specified time allowance. The results further suggest that the AGRS procedure outperforms the traditional local search procedure because it enhances solution diversification. A complete hypothetical case study is conducted to compare the dynamic solutions given by our decision support system with fixed expert decisions. The dynamic solutions perform better than the fixed solutions in terms of multiple objectives. This further demonstrates that the evolutionary decision procedure can make decisions suitable for changing situations. Through the case study, we also gain an important insight into the disaster response problem. The response system can function more efficiently by reducing the congestion at the scene as a consequence of dispatching fewer but adequate responders.

As a short concluding remark, the work presented in this dissertation makes a significant contribution to the simulation modeling and optimization research as well as the disaster response studies.

7.2 FUTURE RESEARCH DIRECTIONS

In this dissertation, an integrated simulation-based decision support system is developed. In the development and testing processes, a number of research issues arose and attracted the author's attention. A number of issues could be future research directions to extend the work further.

The hybrid agent-based discrete event simulation framework successfully separates agent rules from simulation and stores the rules in an independent rule-based system. In the current implementation, a single fixed rule base is used and all the rules are in the deterministic "IF-THEN" format. For complex systems, multiple operational rule sets could be applied to the agents (entities) in different situations. For example, in the disaster response system, the responder agents could comply with national response protocols or follow local contingency plans. The simulation system should be able to flexibly run both policies and assess their performance. Thus, a significant extension to the hybrid framework would be the development of flexible schemes to store changeable rule sets for different scenarios. Database inquiries could be a potential direction to implement this idea. Agent rules' conditions and consequences are stored numerically in a database which is interfaced with the simulator. Agents' decisions can be obtained by logically inquiring the rule database iteratively. Such a database is easily changeable, as is the rule set.

In addition, a probability should be attached to each rule segment. Under the same condition, there could be multiple alternative actions for an agent and it might choose one based on the probabilities. This implementation would model the agents to be more intelligent and realistic because in reality, not only one consequence occurs under a situation. To build more intelligent agents, we might consider utilizing the simulation's situation awareness capability to let agents autonomously decide the best rule to execute based on future predictions.

The evolutionary decision making procedure developed in this dissertation addresses the dynamic nature of complex decisions. In the procedure, decisions are only optimized for a short time horizon because long-term planning might neglect the change in the system and misrepresent the actual situations. The next question would be when is the optimal point of time to make a new decision? This question should be answered by another research investigation – decision interval optimization. In the current work, the decision interval is mainly fixed in length although a primary result about the optimal length of decision evaluation horizon is shown through experiments.

One advantage of simulations is that a large number of data can be collected through extensive experiments although the runs require a considerable amount of computer time. How to utilize offline experimental data to search for optimal solutions online is another potential research topic. The offline data could be used to construct better analytical estimation models and/or guide local search to converge to global optimum more quickly. A tradeoff exists between using analytical and simulation models. Given that the analytical model has a good prediction capability of the corresponding simulation results, the potential exists for eliminating expensive simulation runs and only using the analytical model to obtain basic insights into various plausible decisions, then choosing the best one. In the AGRS procedure, we perturb each of the analytical objective functions with a randomly distributed error factor to estimate the corresponding simulation results. However, different objectives have different contributions to the error. Identifying the key sources of errors and imposing more perturbations on these factors would be also promising in order to construct better analytical estimations to the simulation.

Disaster response planning and management is a very complex and large-scale problem that involves many factors. With respect to the disaster simulation system, significant research work could be done to extend the model and address the issues such as multi-scene response planning, post-hospital transfer and operations, and traffic interferences of regular and emergency vehicles. For such a complex problem, a large number of open questions are left for the successor researchers to complete.

7.3 A PROMISING EXTENSION TO AGRS

Upon the conclusion of this dissertation, a new avenue has been suggested that offers great potential for improving the current Analytically Guided Randomized Search (AGRS) algorithm, which has been described in section 5.5.

Recall that the current AGRS constructs and solves a number of approximate analytical optimization models by perturbing the original, basic analytical model, simulates all of the analytical solutions and chooses the best solution based on the simulation results. This procedure is illustrated in Figure 7-1 as a part of Figure 5-4.

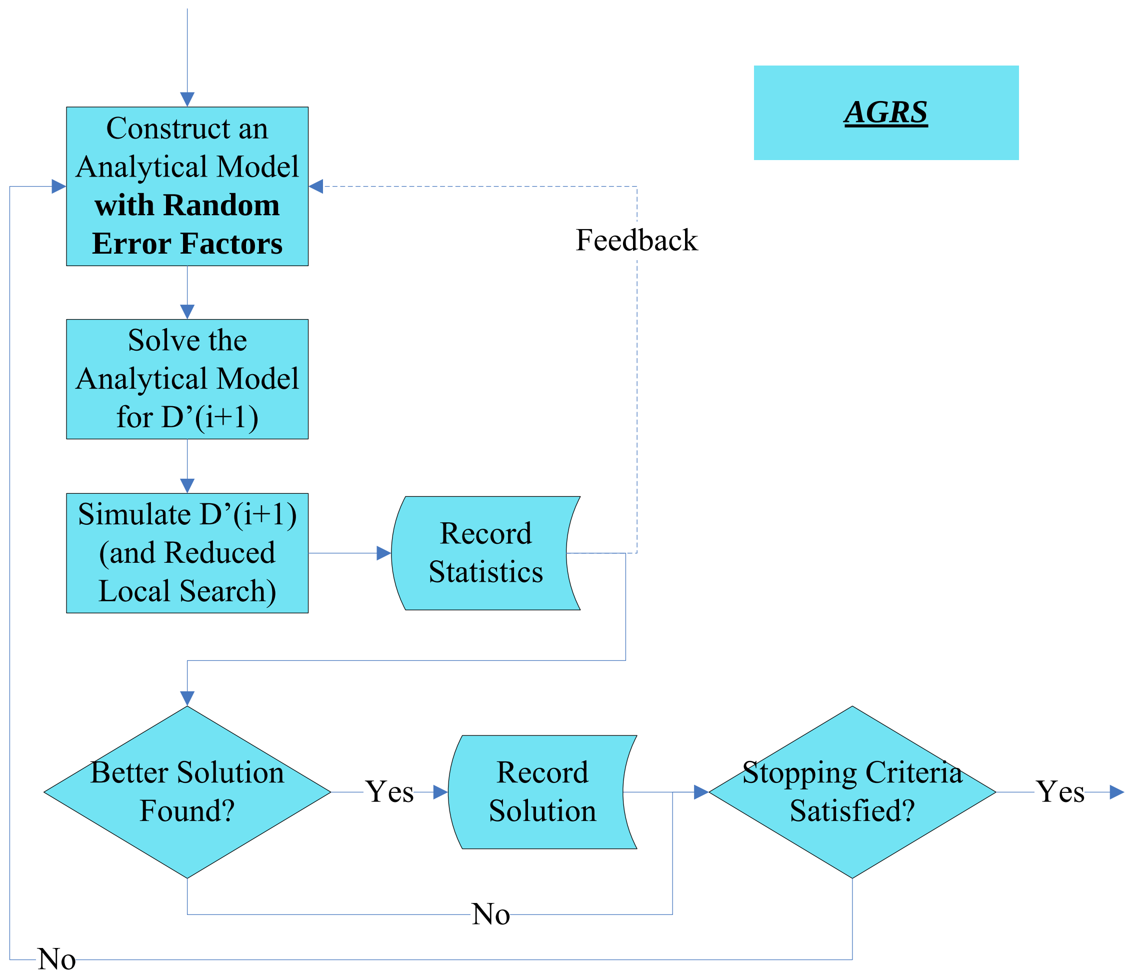

Figure 7-1. Analytically Guided Randomized Search (AGRS)

The above implementation of AGRS incorporates the random differences between approximate analytical models and actual simulation results in order to diversify the solution candidates more rationally than the traditional local search does. However, in each iteration, only one candidate solution is optimized by a perturbed analytical model and then this solution is evaluated by the simulator. Since simulation runs consume a considerable amount of computation time, in this case, only a few solutions can be searched within a limited time frame.

From the preliminary system study in subsection 6.3.2, Figure 6-9 indicates a strong positive correlation between the original analytical model results and corresponding simulation results. In other words, if a solution is better for the analytical model, there is a good chance that this solution is better for the simulation model. This relationship is further verified by a linear regression study on the data obtained from the results in subsection 6.3.3.3. The regression for about 1500 simulation results vs. the original analytical model data suggests a very high R^2 value of 0.958. All these facts appoint to significant, promising improvement of the current AGRS algorithm for our situation. Because solving analytical optimization models are relatively more efficient than running the simulation model, we could then choose to construct and solve more analytical models with perturbations and then simulate only several of the best candidate solutions using the simulator. The detailed algorithm is presented as follows:

Step 1. Construct n' analytical models with perturbations (including the one without perturbation, i.e., the original model) and solve for n solutions. Note that $n' \geq n$ because different analytical models may produce an identical solution.

Step 2. Quickly evaluate the n solutions using the original analytical objective function and pick the best m out of n candidates based on the analytical evaluations. The tradeoff between n and m is an interesting decision here. Within a fixed time span, an increase in n necessarily leads to a decrease in m, and vice versa, because solving an analytical model and running a simulation both take time although they have different computational costs. For example, suppose it takes 20 seconds to solve an analytical optimization model and it takes 5 minutes to run a simulation model. Regardless of other minor computation time, we have to reduce one simulation run in order to solve for 15 more analytical solutions within the same amount of time.

Step 3. Simulate the m candidate solutions from Step 2 and choose the best one based on the simulation results.

The above modified procedure is more promising than the previously described AGRS algorithm in the sense that it further diversifies the solution pool by reallocating some of the computation effort from the expensive simulation runs to solve more analytical models. However, the effectiveness of the new procedure requires that a strong positive correlation between the original analytical model and the simulation model exists; in other words, the analytical model should be able to accurately predict the trend of the simulation model. Extensive experimental data are needed to further validate this method. The work of implementation and validation will be left for future research.

BIBLIOGRAPHY

1. Andradottir, S. 1998. **A review of simulation optimization techniques**. In the *Proceedings of the 1998 Winter Simulation Conference*, 151-8.

2. Arizona State University. 2005. **Who's on first? Decision-making in the midst of disaster**. http://knowledge.wpcarey.asu.edu/index.cfm?fa=viewfeature&id=1111 (accessed September 7, 2006).

3. Barton, R.R. 1998. **Simulation metamodels**. In the *Proceedings of the 1998 Winter Simulation Conference*, 167-76.

4. Banks, J., Carson II, J.S., Nelson, B.L., and Nicol, D.M. 2005. ***Discrete-event System Simulation***. Upper Saddle River, N.J. : Pearson Prentice Hall.

5. Van Beers, W.C.M. 2005. **Kriging metamodeling in discrete-event simulation: An overview**. In the *Proceedings of the 2005 Winter Simulation Conference*, 202-8.

6. Blackwell, T.H. and Kaufman, J.S. 2002. **Response time effectiveness: Comparison of response time and survival in an urban emergency medical services system**. *Academic Emergency Medicine*, 9(4):288-95.

7. Blanchard, B.S. 1992. ***Logistics Engineering and Management***, 4th edition. Englewood Cliffs, N.J. : Prentice-Hall.

8. Bolstad, P. 2002. ***GIS Fundamentals: A First Text on Geographic Information Systems***. White Bear Lake, Minn. : Eider Press.

9. Bonabeau, E. 2002. **Agent-based modeling: Methods and techniques for simulating human systems**. In the *Proceedings of the National Academy of Sciences*, 99(3):7280-87.

10. Born, R.G. 2005. **Teaming discrete-event simulation and geographic information systems to solve a temporal/spatial business problem**. In the *Proceedings of the 2005 Winter Simulation Conference*, 2482-91.

11. Brekelmans, R., Driessenb, L., Hamersc, H., and Hertogc, D. 2005. **Constrained optimization involving expensive function evaluations: A sequential approach**. *European Journal of Operational Research*, 160:121-38.

12. Carley, K.M., Fridsma, D.B., and Casman E., Yahja, A., Altman, N., Chen, L.-C., Kaminsky, B., and Nave, D. 2006. **BioWar: Scalable agent-based model for bioattacks**. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 36(2):252-65.

13. Cassandras, C.G. and Lygeros, J. (eds.) 2007. ***Stochastic Hybrid Systems***. Boca Raton, Fla. : CRC/Taylor & Francis.

14. Cerny, V. 1985. **Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm**. *Journal of Optimization Theory and Applications*, 45(1):41-51.

15. Cheng, Y. 1998. **Rule-based train traffic reactive simulation model**. *Applied Artificial Intelligence*, 12(1):5-27.

16. Collett, D. 2003. ***Modelling Survival Data in Medical Research***. Boca Raton, Fla. : Chapman & Hall/CRC.

17. Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. 2001. ***Introduction to Algorithms***, 2nd edition. Cambridge, Mass. : MIT Press.

18. Crescenzi, P. and Kann, V. (eds.) ***A Compendium of NP Optimization Problems***. http://www.ensta.fr/~diam/ro/online/viggo_wwwcompendium/node276.html (accessed January 23, 2008)

19. Cutter, S.L. and Gall, M. 2007. **Hurricane Katrina: A Failure of Planning or a Planned Failure?** In C. Felgentreff and T. Glade (eds.), *Naturrisiken und Sozialkatastrophen*, 1st edition. Germany : Spektrum Akademischer Verlag/Springer.

20. Department of Homeland Security (DHS). 2007. ***National Preparedness Guidelines***. September 2007, Washington, DC.

21. Ebeling, C.E. 1997. ***An Introduction to Reliability and Maintainability Engineering***. Boston, Mass. : McGraw-Hill.

22. Ehrgott, M. 2005. ***Multicriteria Optimization***, 2nd edition. Berlin; New York : Springer.

23. Elandt-Johnson, R. and Johnson, N. 1980. ***Survival Models and Data Analysis***. New York, N.Y. : John Wiley & Sons.

24. Emergency & Disaster Management Inc. **Hazards and disasters**. http://www.emergency-management.net/disaster.htm (accessed August 18, 2006).

25. Endsley, M.R. 1988. **Situation awareness global assessment technique (SAGAT)**. In the *Proceedings of the National Aerospace and Electronics Conference* (NAECON), 789-95.

26. Endsley, M.R. 1995. **Toward a theory of situation awareness in dynamic systems**. *Human Factors*, 37(1):32-64.

27. Endsley, M.R. 2000. **Theoretical underpinnings of situation awareness: A critical review**. In M.R. Endsley and D.J. Garland (eds.), *Situation Awareness Analysis and Measurement*, Mahwah, N.J. : LEA.

28. Epstein, J.M. and Axtell, R. 1996. ***Growing Artificial Societies: Social Science From the Bottom Up***. Washington, D.C. : Brookings Institution.

29. Erkin, Z., Wu, S., Shuman, L., and Bidanda, B. 2007. **Locating the agents with consideration of homogenous workload**. 2007 INFORMS Annual Meeting, Seattle, WA.

30. ESRI. 2006. **New Orleans, Louisiana, leverages spatial data to aid in damage assessment efforts after Hurricane Katrina**. http://www.esri.com/news/arcnews/spring06articles/new-orleans.html (accessed October 13, 2006).

31. Feo, T.A. and Resende, M.G.C. 1995. **Greedy randomized adaptive search procedures**. *Journal of Global Optimization*, 6:109-33.

32. Finlay, P.N. and Wilson J.M. 1987. **The paucity of model validation in operational research projects**. *The Journal of the Operational Research Society*, 38(4):303-8.

33. Fu, M.C. 2002. **Optimization for simulation: theory vs. practice**. *INFORMS Journal on Computing*, 14(3):192-215.

34. Full, Robert A., Allegheny County Chief of Emergency Services. 2007. **Personal communication by Ken Sochats**.

35. Gambardella, L.M., Rizzoli, A.E., and Funk, P. 2002 **Agent-based planning and simulation of combined rail/road transport**. *Simulation*, 78(5):293-303.

36. Garey, M.R. and Johnson, D.S. 1979. ***Computers and Intractability : A Guide to the Theory of NP-completeness***. San Francisco, Calif. : W. H. Freeman.

37. Gass, S.I. 1983. **Decision-aiding models: Validation, assessment, and related issues for policy analysis**. *Operations Research*, 31(4):603-1.

38. Glover, F. 1977. **Heuristics for integer programming using surrogate constraints**. *Decision Sciences*, 8:156–66.

39. Glover, F. 1989. **Tabu search – Part I**. *ORSA Journal on Computing*, 1(3):190-206.

40. Glover, F. 1990. **Tabu search – Part II**. *ORSA Journal on Computing*, 2(1):4-32.

41. Glover, F. 1994. **Genetic algorithms and scatter search: unsuspected potentials**. *Statistics and Computing*, 4:131-40.

42. Glover F., Kelly, J.P., and Laguna, M.L. 1996. **New advances and applications of combining simulation and optimization**. In the *Proceedings of the 1996 Winter Simulation Conference*, 144–52.

43. Glover, F. and Laguna, M. 1997. ***Tabu Search***. Boston, Mass. : Kluwer Academic Publishers.

44. Glover, F., Kelly, J.P. and Laguna, M. 1999. **New advances for wedding optimization and simulation**. In the *Proceedings of the 1999 Winter Simulation Conference*, 255-60.

45. Glover, F.W. and Kochenberger, G.A.. 2003. ***Handbook of Metaheuristics***, 1st edition. Norwell, Mass. : Kluwer Academic.

46. GNU Operating System. **GLPK (GNU Linear Programming Kit)**. http://www.gnu.org/software/glpk/ (accessed June 7, 2008).

47. Goldberg, J., Dietrich, R., Chen, J.M., Valenzuela, T., and Criss, E. 1990. **A simulation model for evaluating a set of emergency vehicle base locations: Development, validation, and usage**. *Socio-Economic Planning Sciences*, 24(2):125-41.

48. Goldberg, J.B. 2004. **Operations research models for the deployment of emergency services vehicles**. *EMS Management Journal*, 1(1):20-39.

49. Green, L. and Kolesar, P. 1989. **Testing the validity of a queueing model of police patrol**. *Management Science*, 35(2):127-48.

50. Gurney, K. 1997. ***An Introduction to Neural Networks***. London : UCL Press.

51. Haddow, G.D. and Bullock, J.A. 2003. ***Introduction to Emergency Management***. Boston, Mass. : Butterworth-Heinemann.

52. Haykin, S. 1999. ***Neural Networks: A Comprehensive Foundation***, 2nd edition. Upper Saddle River, N.J. : Prentice Hall.

53. Holland, J.H. 1992. ***Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence***, 1st edition. Cambridge, Mass. : MIT Press.

54. Hranac, R., Sterzin, E., Krechmer, D., Rakha, H., and Farzaneh, M. 2006. **Empirical studies on traffic flow in inclement weather**. *U.S. DOT Federal Highway Administration Technical Report*, Publication No. FHWA-HOP-07-073.

55. Ignizio, J.P. 1991. ***An Introduction to Expert Systems: The Development and Implementation of Rule-based Expert Systems***. New York, N.Y. : McGraw-Hill.

56. ILOG Inc. 2003. ***ILOG CPLEX 9.0 User's Manual***.

57. Kachani, S. and Perakis, G. 2001. **Second-order fluid dynamics models for travel times in dynamic transportation networks**. In the *Proceedings of the 2001 IEEE Intelligent Transportation Systems Conference*, 251-6.

58. Kelton, W.D., Sadowski, R.P., and Sturrock, D.T. 2007. ***Simulation with Arena***, 4th edition. New York, N.Y. : McGraw-Hill.

59. Kilmer, R.A. 1994. ***Artificial Neural Network Metamodels of Stochastic Computer Simulations***. Pittsburgh, Pa. : University of Pittsburgh Doctoral Dissertation.

60. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. 1983. **Optimization by simulated annealing**. *Science*, New Series, 220(4598):671-80.

61. Klein, G.A. 1993. **A recognition-primed decision (RPD) model of rapid decision making**. In Klein, G.A., Orasanu, J., Calderwood, R., and Zsambok, C.E. (eds.), *Decision Making in Action: Models and Methods*, Norwood, N.J. : Ablex Publishing Co.

62. Lai, J.-P. 2006. ***Surrogate Search: A Simulation Optimization Methodology for Large-scale Systems***. Pittsburgh, Pa. : University of Pittsburgh Doctoral Dissertation.

63. Larson, R.C. 1974. **A hypercube queuing model for facility location and redistricting in urban emergency services**. *Computers and Operations Research*, 1(1):67-95.

64. Larson, R.C. 1975. **Approximating the performance of urban emergency service systems**. *Operations Research*, 23(5):845-68.

65. Law, A.M. and Kelton, W.D. 2000. ***Simulation Modeling and Analysis***, 3rd edition. Boston, Mass. : McGraw-Hill.

66. Lee, Bruce Y., Assistant Professor of Medicine and Biomedical Informatics, University of Pittsburgh, Pittsburgh, Pa. November 2007. **Personal communication**.

67. Lee, E.K., Maheshwary, S., Mason, J., and Glisson, W. 2006. **Large-scale dispensing for emergency response to bioterrorism and infectious-disease outbreak**. *Interfaces*, 36(6):591-607.

68. Lee, S., Pritchett, A., and Goldsman, D. 2001. **Hybrid agent-based simulation for analyzing the national airspace system**. In the *Proceedings of the 2001 Winter Simulation Conference*, 1029-36.

69. Lerner, E.B., Billittier IV, A.J., Dorn, J.M., and Wu, Y.-W.B. 2003. **Is total out-of-hospital time a significant predictor of trauma patient mortality?** *Academic Emergency Medicine*, 10(9):949-54.

70. Li, H.-L. 1994. **A global approach for general 0–1 fractional programming**. *European Journal of Operational Research*, 73(3):590–96.

71. Liu, Y., Zou, N., and Chang, G.-L. 2005. **An integrated emergency evacuation system for real-time operations – A case study of Ocean City, Maryland under hurricane attacks**. In the *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, 281-6.

72. Liu, Y., Lai, X. and Chang, G.-L. 2006. **Two-level integrated optimization system for planning of emergency evacuation**. *Journal of Transportation Engineering*, 132(10):800-7.

73. Mahmassani, H.S., Qin, X., and Zhou, X. 2004. ***DYNASMART-X Evaluation for Real-time TMC Application: Irvine Test Bed***. College Park, Md. : Maryland Transportation Initiative.

74. Mason, A., Day, P., Henderson, S.G., Meyer, J., Snowdon, J., and Waite, J. 2003. **Development of a simulation package for modeling emergency medical service operations**. In the *Proceedings of the 8th Annual International Conference on Industrial Engineering Theory, Applications and Practice*, 556-9.

75. Mason, Andrew J., Senior Lecturer, Department of Engineering Science, School of Engineering, University of Auckland, Auckland, New Zealand. August 2007. **Personal communication**.

76. Masri, A. and Moore II, J.E. 1993. **Integrated planning information systems: Context, design requirements, and prospects**. *Computers, Environment and Urban Systems*, 17(6):491-511.

77. Masri, A. and Moore II, J.E. 1995. **Integrated planning information systems: Disaster planning analysis**. *Journal of Urban Planning and Development*, 121(1):19-39.

78. Mendonca, F.C. and Morabito, R. 2001. **Analysing emergency medical service ambulance deployment on a Brazilian highway using the hypercube model**. *Journal of the Operational Research Society*, 52(3):261-70.

79. Merriam-Webster Online. http://www.webster.com (accessed December 5, 2007).

80. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. 1953. **Equation of state calculations by fast computing machines**. *Journal of Chemical Physics*, 21(6):1087-92.

81. Montgomery, D.C., Peck, E.A., and Vining, G.G. 2001. ***Introduction to Linear Regression Analysis***, 3rd edition. New York, N.Y. : Wiley-Interscience.

82. National Fire Protection Association (NFPA). 2002. **NFPA 1561: Standard on emergency services incident management system**. Quincy, Mass. : NFPA.

83. Nemhauser, G.L. and Wolsey, L.A. 1999. ***Integer and Combinatorial Optimization***. New York, N.Y. : Wiley and Sons.

84. New York Centers for Terrorism Preparedness and Planning (NYCTP). 2006. ***Mass Casualty/Trauma Event Protocol***, draft. New York, N.Y. : NYCTP, July 2006.

85. New York State Department of Health. **EMS mutual aid planning guidelines**. http://www.health.state.ny.us/nysdoh/ems/policy/89-02.htm (accessed April 8, 2008).

86. Nikoukaran, J. and Paul, R.J. 1999. **Software selection for simulation in manufacturing: A review**. *Simulation Practice and Theory*, 7(1):1-14.

87. Nossiter, A. 2006. **New Orleans population is reduced nearly 60%**. *New York Times*, October 7, 2006. http://www.nytimes.com/2006/10/07/us/07population.html (accessed October 16, 2006).

88. Office of Emergency Management, City & County of Boulder, Colorado. **Emergency operations plan**. http://boulderoem.com/plan.html (accessed April 7, 2008).

89. OR/MS Today. 2005. **Simulation software survey**. http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html (accessed September 19, 2007).

90. Pardalos, P.M. and Wolkowicz, H. (eds.) 1994. ***Quadratic Assignment and Related Problems : DIMACS Workshop, May 20-21, 1993***. Providence, R.I. : American Mathematical Society.

91. Peral-Gutierrez de Ceballos, J., Turégano-Fuentes, F., Perez-Diaz, D., Sanz-Sanchez, M., Martin-Llorente, C., and Guerrero-Sanz, J.E. 2005. **11 March 2004: The terrorist bomb explosions in Madrid, Spain – an analysis of the logistics, injuries sustained and clinical management of casualties treated at the closest hospital**. *Critical Care*, 9:104-11.

92. Pichitlamken, J. and Nelson, B.L. 2002. **A combined procedure for optimization via simulation**. In the *Proceedings of the 2002 Winter Simulation Conference*, 292-300.

93. Pichitlamken, J. and Nelson, B.L. 2003. **A combined procedure for optimization via simulation**. *ACM Transactions on Modeling and Computer Simulation*, 13(2):155-79.

94. Prokopyev, O.A., Meneses, C., Oliveira, C.A.S., and Pardalos, P.M. 2005. **On multiple-ratio hyperbolic 0-1 programming problems**. *Pacific Journal of Optimization*, 1(2):327-45.

95. Reeves, C.R. (editor) 1993. ***Modern Heuristic Techniques for Combinatorial Problems***. New York, N.Y. : Halsted Press.

96. Robinson, S. 2003. ***Simulation – The Practice of Model Development and Use***. Hoboken, N.J. : John Wiley.

97. Rockwell Software. 2005. ***Arena User's Guide***.

98. Russell, S. and Norvig, P. 2003. ***Artificial Intelligence – A Modern Approach***, 2nd edition. Upper Saddle River, N.J. : Prentice Hall/Pearson Education.

99. Sacco, W.J., Navin, D.M., Fiedler, K.E., Waddell II, R.K., Long, W.B., and Buckman Jr., R.F. 2005. **Precise formulation and evidence-based application of resource-constrained triage**. *Academic Emergency Medicine*, 12(8):759-70.

100. Sherali, H.D., Adams, W.P. 1994. **A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems**. *Discrete Applied Mathematics*, 52:83-106.

101. Sherali, H.D. 2007. **RLT: A unified approach for discrete and continuous nonconvex optimization**. *Annals of Operations Research*, 149:185-93.

102. Shuman, L., Wolfe, H., Ames IV, J., Etschmaier, M., and Sepulveda, J. 1981. **RURALSIM: A simulation model for designing and evaluating rural EMS systems**. In C. Tilquin (editor), *Systems Science in Health Care*, Toronto : Pergamon Press, 2:961-73.

103. Shuman, L.J., Sepulveda, J., Wolfe, H., and Esposito, G. 1985. **Computer modeling of rural emergency medical services delivery systems**. *Journal of World Association for Disaster and Emergency Medicine*, 1(supplement I):44-8.

104. Shuman, L.J., Wolfe, H., and Gunter, M.J. 1992. **RURALSIM: The design and implementation of a rural EMS simulator**. *Journal of the Society for Health Systems*, 3(3):54-71.

105. Shuman, L.J., et al. **RURALSIM manual, code, and data**. RURALSIM was developed under Department of Transportation Contract DOT-8S-8-02028.

106. Sochats, K., Director of Center for National Preparedness, Director of Visual Information Systems Center, University of Pittsburgh, Pittsburgh, Pa. October 2007. **Personal communication**.

107. Son, Y.J., Wysk, R.A., and Jones, A.T. 2003. **Simulation-based shop floor control: Formal model, model generation and control interface**. *IIE Transactions*, 35(1):29-48.

108. Sridhar, P., Sheikh-Bahaei, S., Xia, S., and Jamshidi, M. 2003. **Multi agent simulation using discrete event and soft-computing methodologies**. In the *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2:1711-16.

109. Storer, R.H., Wu, S.D., and Vaccari, R. 1992. **New search spaces for sequencing problems with application to job shop scheduling**. *Management Science*, 38(10):1495-509.

110. Stoy, Walt Alan, Administrative Director, UPMC/HMC Partnership – Qatar Strategic Business Initiatives – International Services; Professor and Director, Emergency Medicine Program, School of Health and Rehabilitation Sciences; Research Professor of Emergency Medicine, School of Medicine, University of Pittsburgh; Director, Office of Education and International Emergency Medicine, Center for Emergency Medicine of Western Pennsylvania, Pittsburgh, Pa. October 2006. **Personal communication**.

111. Suyama, Joe, Department of Emergency Medicine, University of Pittsburgh. October 2006. **Personal communication**.

112. Swisher, J.R., Jacobson, S.H., and Yucesan, E. 2003. **Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey**. *ACM Transactions on Modeling and Computer Simulation*, 13(2):134-54.

113. Swisher, J.R., Hyden, P.D., Jacobson, S.H., and Schruben, L.W. 2004. **A survey of recent advances in discrete input parameter discrete-event simulation optimization**. *IIE Transactions*, 36(6):591-600.

114. Takeda, R.A., Widmer, J.A., and Morabito, R. 2007. **Analysis of ambulance decentralization in an urban emergency medical service using the hypercube queueing model**. *Computers and Operations Research*, 34(3):727-41.

115. Tawarmalani, M., Ahmed, S., and Sahinidis, N.V. 2002. **Global optimization of 0-1 hyperbolic programs**. *Journal of Global Optimization*, 24:385-416.

116. Tekin, E. and Sabuncuoglu, I. 2004. **Simulation optimization: A comprehensive review on theory and applications**. *IIE Transactions*, 36(11):1067-81.

117. Trunkey, D.D. 1983. **Trauma**. *Scientific American*, 249:28-35.

118. Vanderbei, R.J. 2001. ***Linear Programming: Foundations and Extensions***, 2nd edition. Boston, Mass. : Kluwer Academic.

119. Wiley, R.B. and Keyser, T.K. 1998. **Discrete event simulation experiments and geographic information systems in congestion management planning**. In the *Proceedings of the 1998 Winter Simulation Conference*, 1087-93.

120. Willis, T. and Newsome, B. 2006. ***Beginning Visual Basic 2005***. Indianapolis, Ind. : Wiley.

121. Wu, S., Shuman, L., Bidanda, B., Kelley, M., Sochats, K., and Balaban, C. 2007a. **Disaster policy optimization: A simulation based approach**. In the *Proceedings of the 2007 Industrial Engineering Research Conference*, 872-7.

122. Wu, S., Shuman, L., Bidanda, B., Kelley, M., Sochats, K., and Balaban, C. 2007b. **Embedding GIS in disaster simulation**. In the *Proceedings of the 27th Annual ESRI International User Conference*, Paper No. UC1847.

123. Wu, S., Shuman, L.J., Bidanda, B., Kelley, M., Sochats, K., and Balaban, C. 2007c. **System implementation issues of Dynamic Discrete Disaster Decision Simulation System (D^4S^2) - Phase I**. In the *Proceedings of the 2007 Winter Simulation Conference*, 1127-34.

124. Wu, S., Shuman, L., Bidanda, B., Kelley, M., Sochats, K., and Balaban, C. 2008a. **Agent-based Discrete Event Simulation Modeling for Disaster Responses**. In the *Proceedings of the 2008 Industrial Engineering Research Conference*, 1908-13.

125. Wu, S., Shuman, L., Bidanda, B., Prokopyev, O., Kelley, M., Sochats, K., and Balaban, C. 2008b. **Simulation-based Decision Support System for Real-time Disaster Response Management**. In the *Proceedings of the 2008 Industrial Engineering Research Conference*, 58-63.

126. Wu, T.-H. 1997. **A Note on a global approach for general 0-1 fractional programming**. *European Journal of Operations Research*, 101:220-23.

127. Yang, Q. and Koutsopoulos, H.N. 1996. **A microscopic traffic simulator for evaluation of dynamic traffic management systems**. *Transportation Research, Part C: Emerging Technologies*, 4(3):113-29.

128. Yi, W. and Ozdamar, L. 2007. **A dynamic logistics coordination model for evacuation and support in disaster response activities**. *European Journal of Operational Research*, 179(3):1177-93.

129. Zhigljavsky, A.A. and Pintér, J. (eds.). 1991. ***Theory of Global Random Search***. Boston, Mass. : Kluwer Academic Publishers.

130. Zsambok, C.E. and Klein, G. (eds.) 1997. ***Naturalistic Decision Making***. Mahwah, N.J. : Lawrence Erlbaum Associates.