A+BE Architecture and the Built environment

#04 2015

# Path planning for first responders in the presence of moving obstacles

Zhiyong Wang

# PATH PLANNING FOR FIRST RESPONDERS IN THE PRESENCE OF MOVING OBSTACLES

# PATH PLANNING FOR FIRST RESPONDERS IN THE PRESENCE OF MOVING OBSTACLES

**Proefschrift**

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 09 juni 2015 om 10:00 uur

door

**Zhiyong WANG**

Master of Engineering, Tongji University, China
geboren te Xingning, Guangdong province, China.

This dissertation has been approved by the

Promotor: Prof. dr. ir. P. J. M. van Oosterom
Copromotor: Dr. S. Zlatanova

Composition of the doctoral committee:

| | |
|---|---|
| Rector Magnificus | voorzitter |
| Prof. dr. ir. P. J. M. van Oosterom | Technische Universiteit Delft, promotor |
| Dr. S. Zlatanova | Technische Universiteit Delft, copromotor |

Independent members:

| | |
|---|---|
| Prof. dr. -ing. L. Meng | Technische Universität München |
| Prof. dr. ir. H. Scholten | Vrije Universiteit Amsterdam |
| Prof. drs. dr. L. J. M. Rothkrantz | Technische Universiteit Delft |
| Prof. dr. ir. P. van Gelder | Technische Universiteit Delft |
| Prof. dr. M. Worboys | University of Greenwich |

This work was supported by the China Scholarship Council.

Published and distributed by: Zhiyong Wang
E-mail: zwang19840102@gmail.com

*Keywords:* Path planning; Taxonomy; Multi-agent system; Spatial data model; Algorithms; Moving obstacles

*Cover image:* Feuerwehr (licensed copy) © by davis (http://Fotolia.com)

*Cover design:* Sirene Ontwerpers



ISBN 978-94-6186-477-2
ISSN 2212-3202

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

*To my beloved parents*
献给我亲爱的父亲和母亲

# ACKNOWLEDGEMENTS

The path across three departments to become a doctor is not easy. It is long, sometimes boring and lonely, and full of various difficulties. But I feel very fortunate that I was not alone on this path, but accompanied by a group of nice people from all over the world. Here I would like to take this opportunity to express my sincere gratitude to those people who provide help and support to my research and make this thesis possible.

First of all, I would like to thank my supervisors, Sisi Zlatanova and Peter van Oosterom. I am grateful for their guidance, patience, comments, suggestions, and ideas that contribute to my work. Without their support, I could never have overcome the difficulties I met.

I would also like to thank Cees Timmers and Franca Post for their support to me throughout my staying in the Netherlands.

I thank Jantien Stoter for her support to my work and introducing me to the 3D world. I also thank Hugo Ledoux for giving me all kinds of help and making enjoyable and pleasant atmosphere in the office.

Many thanks go to all other colleagues and visitors. It is my great privilege to meet you and work with you in NL. I really appreciate all of your assistance whenever I need them. Ken and Ravi, your time and assistance in the preparation of my thesis are highly appreciated. Liu, Filip, Radan, Pirouz thank you for sharing your ideas and thoughts and accompanying me in this path towards a doctor. Wish all of you success in your PhD. Martijn, Theo, Tjeu, Wilko, Marian, and Edward thank you for your kind support, nice conversations, and inspiring discussions. A special word of gratitude goes to Elfriede for helping me deal with all sorts of administrative issues. Amin, Filippo, Davide, I really enjoy your companionship in the same office. I thank our former Chinese visitors, Shen Ying, Hua Liu, Lei Niu, Junqiao Zhao, Lina Huang, Yeting Zhang, and Yan Zhou, Youhui Wang, Zheng Gong for their suggestions as well as the delicious dinners. Serpil and Elif, thank you for buying me coffee and hope you success in your projects.

I also want to thank Aitor Moreno and Carlos Toro for providing the data and hosting me at Vicomtech-IK4, and thank Mathieu, Paddy, and Wim for their help in applying for the projects and funds. Special thanks to Claire Taylor for helping me correct the English text in this thesis.

Here I want to express my gratitude to my Chinese friends who make my life in NL full of joy and support: Jia Xu, Yongjia Li, Jianbin Xin, Qi Tu, Wenjing Deng, Ni Kang and Junchao Xu, Chao Qu and Yun Ling, Yangyang Shi, Zhenpei Wang, Cuiting Chen, Changyun Wei, Tingting Zhang, Jie Jiang and Wen Li. Furthermore, I thank my former roommates, Li Mo, Yong Zhang, Song Yang, Yunlong Gong, Bin Zhao, Yue Qiu for sharing a cozy home with me. Jiayuan Zhang and Da Li thank you for your friendship and support over the past eighteen years. Particularly, I would like to thank Chang Wang, Yi Sao, Ran Shang and Jingyi Hu for their finical support that makes me survive when I was having some economic difficulties.

My thanks go to the secretaries of OTB and the secretaries of Urbanism for their great administrative support and providing a pleasant working environment.

I sincerely appreciate China Scholarship Council (CSC) for funding my studying abroad.

Although it is impossible to mention all the people here, still I would like to thank all the readers of my thesis for their very valuable and constructive suggestions, and thank the people who contributed either directly or indirectly to my publications and my PhD research.

Last but not least, I want to give my biggest thanks to my parents Yunxin Wang and Lisha Zhang, and my relatives for their unconditional love to me. Especially when I face difficulties and doubts, they always support me, giving me encouragement, strength, confidence and wisdom.

To all of you, thank you very much!

*Zhiyong WANG*
*18th May 2015*

## TABLE OF CONTENTS

# 1 INTRODUCTION

This chapter gives an introduction to the PhD research conducted in the thesis. It first presents the motivation of the research by showing the research problem, the scientific gap, and possible solutions to the problems (see Section 1.1). Then it brings forward the main research question, and the sub-research questions that need to be answered to achieve the intended objective (see Section 1.2). Section 1.3 highlights the research topics that are inside and outside the scope of this reach. In Section 1.4, we present the research methodology that leads us to the answers to those research questions. Section 1.5 provides a list of the tools and data that are used throughout the development of the prototype system. At the end of this chapter, we give an overview of this thesis (see Section 1.6).

## 1.1 Research motivation

Disaster relief involves a number of coordinated activities including searching and rescuing survivors, health and medical assistance, food and water distribution, and transporting injuries. Much of the successful and effective relief work relies on the safe and fast navigation. Therefore, route planning during disasters plays an important role in the disaster response phase and has attracted an increasing interest in the navigation field.

The complexity of emergency procedures and dynamics of transportation network affected by disasters pose a set of serious challenges to technology innovations related to Location-Based Service (LBS) and geographic information systems (GIS). One of challenging issues in major disasters is that multiple responders from different emergency management sectors are involved and need to be navigated. In the Netherlands, the disasters are managed by Processes, which are legally within the Dutch Law for Disasters and Large Accidents (WRZO, http://wetten.overheid.nl/). There are 25 types of Processes that have been formalized for 4 primary Sectors: municipality, fire brigade, police and medical care (Diehl et al., 2006; Zlatanova, 2010). For each process, response

Figure 1.1: Emergency managers in the command and control center (CCC)

teams from different sectors are involved and perform certain tasks. Most of these tasks are associated with locations and consist of a set of operations that should be performed at those locations. For example, in the case of a major flood, the Process *MedicalAid* is started, and ambulances from hospitals have to be sent to different disaster sites to give medical assistance to the affected people. To coordinate efforts towards completion of these tasks, emergency managers (Figure 1.1) have to ensure an efficient allocation of these tasks among a group of first response teams. Because response teams may visit more than one location during a trip, their routes also have to be optimized and coordinated, which makes the navigation problem more complex.

Another important issue is that natural or man-made disasters can create all sorts of moving obstacles (e.g. floods, plumes, fires), which make parts of the road network dangerous to pass through for certain periods of time. For instance, in the context of a fire incident that results in moving contaminant plumes (see Figure 1.2), these moving plumes have harmful effects on human health, and can be considered as obstacles that influence the availabilities of some roads. Figure 1.3 presents an example of a moving obstacle affecting a road segment $A\_B$ connected by two junctions $A$ and $B$. As shown in the figure, the obstacle moves and intersects the road segment $A\_B$ during the temporal interval $[t_2, t_3]$. When the emergency response units arrive at a junction of the road segment before time $t_2$, different situations can be distinguished. They can either wait until the road is open again, or choose an alternative route, or even go through the plumes if they have protective suit against toxic materials, all of which increases the difficulty of routing among the moving obstacles.

Figure 1.2: An example of a fire incident (from Zelle et al. (2013))

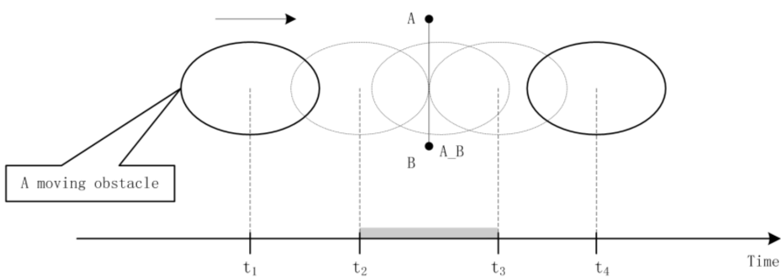



Figure 1.3: Example of a block, and its corresponding temporal intervals when the road segment $A\_B$ is blocked by a moving obstacle

To be able to navigate responders in the presence of moving obstacles, emergency managers may need to know the movement of obstacles caused by disasters and the spatio-temporal information of blocks in the road network. Therefore, an approach is needed to obtain the predicted information of obstacles to support path planning for first responders.

Computer models and simulations have been applied in a variety of fields for studying and predicting the behaviors of complex systems, such as risk management (Ale, 2013; Ale et al., 2014), serious games (Lukosch et al., 2012, 2014), pedestrian modeling (Duives et al., 2013; Greenwood et al., 2014), and traffic management (Hoogendoorn et al., 2014; Knoop and Hoogendoorn, 2014). Especially in disaster management, hazard modeling and simulations provide valuable information about dynamics of hazards (e.g., wildfires, floods, and plumes), and have been used to provide predictions in order to assist decision making in all phases of disaster management. However, traditional computer simulations make little usage of dynamic data and generate simulation results that are certainly different from real hazards, which limits their applications in real disaster response.

With the advance of sensor and network technologies, a new concept called Dynamic Data Driven Applications System (DDDAS) was proposed by Darema (2004) for improving the dynamic modeling and simulations. The DDDAS methodology seeks to integrate the simulation system with real-time measurements to increase the accuracy of simulation results. It allows the system to inject dynamic data to continuously adjust the model, using methods such as data assimilation, Bayesian analysis, sensitivity analysis. Moreover, it also involves use of dynamic data inputs to dynamically control the data collection processes with adaptive sampling techniques. Based on the concept of DDDAS, some researchers have developed a rich set of hazard simulation models which can be driven by dynamic data collected from the field and make more accurate predictions of the hazard spread (Hu, 2011; Rodríguez et al., 2009; Chaturvedi et al., 2005; Mandel et al., 2007). These models simulate the evolution of hazards, which can provide valuable predicted information of the environment affected by disasters, supporting navigation among moving obstacles.

Although the DDDAS-based simulation models show potentials to be used by navigation systems for guiding responders among moving obstacles, there are still a number of problems that arise and remain to be addressed. First of all, the hazard simulation models use different formats in their output. To be able to be handled by GIS tools for spatial analysis, the simulation results from these models need be converted into standard GIS formats. Second, further analysis of data of hazards with the data of road networks is needed to obtain spatio-temporal information for routing, for example, which junctions are closed, which roads will be unavailable, and when they will be blocked. Because the hazard simulations usually generate large amounts of data, the system should be able to fast and automatically process these data to provide real-time navigation support. Third, as successful navigation largely depends on proper access to the relevant information, there is a need for structuring the information for navigation in an efficient manner to rapidly feed the system with the consistent data. Forth, to provide safe and fast routes, a route calculation method should be developed to handle the spatio-temporal information of road networks. Last but not the least, because of the existence of errors in various data sources, the uncertainties would be involved in the hazard simulation results. With some protective equipment, the responders may accept certain levels of uncertainty during their route determination. Therefore, dealing with these uncertainties is also required in the routing process.

## 1.2 Objective and Research questions

Considering a great need for navigation support in the spatio-temporal road network populated by moving obstacles, we can formulate our main research question as follows:

*How do we safely and efficiently navigate one or more first responders to one or more destinations avoiding moving obstacles?*

From the main research question, several sub-questions are derived to conduct a more detailed study of the research topic.

Route planning in disasters is an important but difficult issue. There are various factors that can influence the routing process, which can lead to a number of emergency navigation cases. To be able to define navigation problems that arise during the disaster response, the first question we will address is:

(1) What navigation cases need to be considered for assisting first responders? (Chapter 3)

Information is one of the most critical aspects in emergency navigation, and should be structured and stored in an efficient way. In this thesis, we will explore the use of geo-database management system (geo-DBMS) to handle spatio-temporal information produced during disasters. This naturally leads to our next sub-research question:

(2) What data models should be developed to support path planning among moving obstacles? (Chapter 4)

As we have selected navigation cases for our research, we need to design a system that can support information processing and analysis for navigation in these considered cases. Because the process of path planning in disasters can be divided into a series of sub-processes, the agent technology is applied to combine and handle these processes. This raises the following sub-research question:

(3) What types of agents are needed to assist path planning among moving obstacles? (Chapter 5)

To provide responders with routes avoiding moving obstacles, special algorithms are needed to deal with the spatio-temporal information of the road network. Therefore we will investigate the following sub-research question:

(4) What algorithms should be developed for path planning among moving obstacles? (Chapter 6)

## 1.3 Research scope and limits

Below is a list of main topics that have been included in the scope of this research:

- Design and development of a navigation system for emergency managers in the command and control center as well as responders on the field.
- Data models to structure information for navigation among moving obstacles
- Graph-based path planning algorithms for avoiding moving obstacles

The following is a list of research topics that are in support of this research, but beyond its scope:

- Modeling of hazards
- Collection of real-time measurements
- Indoor navigation
- Position technologies
- 3D visualization
- Path planning in free space
- Use of Volunteered Geographic Information (VGI)
- Implementation on mobile device for navigation
- Human-computer interaction to improve user interface
- Modeling and simulation of the movement of crowds
- Communication between the command and control center and responders

## 1.4 Research methodology

In order to answer the research questions, we carry our this research using a design science research methodology in line with Peffers et al. (2007). In this thesis, the following four methods are employed: literature review, conceptual analysis and design, implementation, assessment and adaptation.

*1. Literature review*

An extensive literature review of publications from crisis management communities (e.g., ISCRAM[1], GI4DM[2], TIEMS[3]) is conducted to define navigation problems that should be considered for this research. Previous works related to multi-agent system, routing algorithms, GIS technology, etc. are also investigated to understand the state-of-the-art in relevant fields. Moreover, technical aspects and implementation issues are taken into consideration during this phase.

*2. Conceptual analysis and design*

In this phase, we conduct a comprehensive analysis of characteristics of the hazards and disaster response process to derive the requirements that the system should meet. Based on the analysis, a conceptual framework is designed to support navigation for first responders. Within this framework, several steps are taken, including definition of the types and functions of the agents, design of data models to structure spatio-temporal data of the road network, and selection of routing algorithms to calculate optimal or near-optimal routes. Furthermore, the existing hazard models are also considered and chosen for the system, providing predictions of the movement of hazards.

*3. Implementation*

This phase of the research aims at implementation of the designed system. The road networks are extracted from available 2D/3D GIS data. Moreover, an approach is developed to transform results from the hazard models to standard GIS data, and to predict effects of hazards on transportation networks. The designed data models are also realized in the database and used in the storage of spatio-temporal data. Besides, different types of agents and the routing algorithms are implemented in the system for spatial processing and analysis.

*4. Assessment and adaptation*

To test the effectiveness of our designed navigation approach in different navigation cases, an agent-based simulation framework is used. The navigation results are evaluated and demonstrated through simulation of dynamic objects (e.g., obstacles, vehicles). Based on observation of agents' behaviors, the developed algorithms are adapted to improve the path finding process. In addition

[1]http://www.iscram.org/
[2]http://www.gi4dm.net/
[3]http://tiems.info/

to that, comparisons of route results (e.g., the route safety, the arrival time of routes, route distance, the total traveling time) between the proposed navigation approach and the existing methods are made to assess the performance of the algorithms.

## 1.5 Related tools and data

Our prototype system is implemented using the following software tools, most of which are open source and free of charge:

1. *Quantum GIS*
   Quantum GIS (QGIS) is a powerful and free open source GIS desktop application that supports GIS data manipulation, operation, analysis, and visualization. It provides various APIs that allow users to integrate it with other GIS packages, such as PostGIS, GRASS, which greatly extend the capabilities of QGIS. More information can be found at `http://www.qgis.org/`.

2. *PostGIS*
   PostGIS is a spatial extension of the PostgreSQL relational database, and supports different spatial data types and functions in compliant with OGC specifications and ISO 19107. It is available at `http://www.postgis.org`.

3. *Enterprise Architect*
   Enterprise Architect is a comprehensive UML modeling and design tool, covering all aspects of the software development cycle. It has been used in many projects, e.g., INSPIRE (Infrastructure for Spatial Information in the European Community). More information can be found at `http://www.sparxsystems.com/products/ea/index.html`.

4. *Java*
   Java is a programming language and computing platform developed by Sun Microsystems. It is one of the most popular platforms for implementing large and long-lived applications. Java is free to download at `http://java.com`.

5. *Eclipse*
   Eclipse is the most widely-used integrated development environment (IDE) for developing Java applications. It is free, open-source, cross-platform, and provides a rich set of plug-ins for various tasks. Available at `http://www.eclipse.org/`.

6. *GeoTools*
   GeoTools is a free Java library and provides powerful tools for handling spatial data. It contains a large number of modules that allow you to analyze, operate and visualize GIS data. Available at `http://www.geotools.org/`.

7. *MASON*
   MASON is a fast discrete-event multi-agent simulation toolkit, designed to be the foundation for large custom-purpose Java simulations. It has various GIS facilities that can be used to integrate and operate GIS data (either raster or vector) directly into the simulation with relative ease. It can be downloaded at `http://cs.gmu.edu/~eclab/projects/mason/`.

   During the last two decades, a variety of Agent Based Modeling tools have been developed, for instance, Repast (Collier, 2003), NetLogo (Wilensky, 1999), MASON (Luke et al., 2003), and SWARM(Minar et al., 1996). Among them, we select Mason for building our system because it has a couple of features that make it suitable for our developments, such as purely in Java, open source, extensive documentation, supporting both 2D and 3D visualization, and providing GIS functionaries. For more information of comparison of ABM tools, readers can refer to Rajendran (2009).

8. *JADE*
   JADE (Java Agent DEvelopment Framework) is a framework for agent development and distributed system. It is developed by TILAB (Telecom Italia LABoratories), in compliance with the FIPA (Foundation for Intelligent Physical Agent) specifications. It can be found at `http://jade.tilab.com`.

   There are a large number of programming languages and development tools that are available for implementation of agent systems, for example, FIPA-OS (Poslad et al., 2000), Jadex (Pokahr et al., 2005), and ZEUS (Lee et al., 1998). In this study, JADE is chosen because it has several advantages compared to other candidates, such as open source, rich APIs, and detailed documentation, and it is widely used by many researchers in both academic and industrial world. More detailed information about comparison between different agent development tools can be found in the following publications: Bordini et al. (2006), López et al. (2010), and Bădică et al. (2011).

9. *OSM2World*
   OSM2World is an open source software that builds three dimensional

models of the environment from OpenStreetMap data. It can be downloaded at `www.osm2world.org`.

For testing of our prototype system, generally there are two kinds of data that are needed:

1. *Data of moving obstacles*
   In this study, we mainly use artificial datasets to simulate the movement of obstacles during disasters. These artificial datasets are created by hand or from existing hazard models. Dozens of datasets have been manually generated to test the performance of our algorithms. We also use a fire simulation model to generate datasets about the predicted spread of the fire, and evaluate the application of our prototype to simulated fire events.

2. *Data of road networks*
   In this research, the OpenStreetMap (www.openstreetmap.org) data is chosen to extract the road network dataset, which can be used for route computation. For visualization, we also use OSM data to provide information on the surroundings, such as houses, gardens, etc., that might not initially be included in the street network model. Besides, other data sources, such as TOP10NL, Navteq, are taken into account in generation of road networks in specific areas.

## 1.6 Outline

Figure 1.4 presents the outline of the thesis. We start with introduction of the reach motivation and questions in Chapter 1. In Chapter 2, we review previous research related to this research. To gain a better understanding of characteristics and differences of navigation problems during disasters, in Chapter 3 we give a taxonomy which classifies different navigation cases with obstacles, and present our proposed approach for addressing these navigation problems. To support the path planning process, a geo-database management system is used in this research. In Chapter 4, a series of data models are designed to structure spatio-temporal information related to disaster response in the database. In Chapter 5, we present our navigation system, which uses the agent technology to support spatial data processing and analysis involved in the path planning among moving obstacles. Chapter 6 shows the algorithms that are used in the system for different types of navigation problems with moving obstacles. In Chapter 7, we apply our multi-agent based navigation system to various navigation cases, and show the results of the applications in different navigation scenarios. At the end of this dissertation, we draw some conclusions and give directions for future work (Chapter 8).

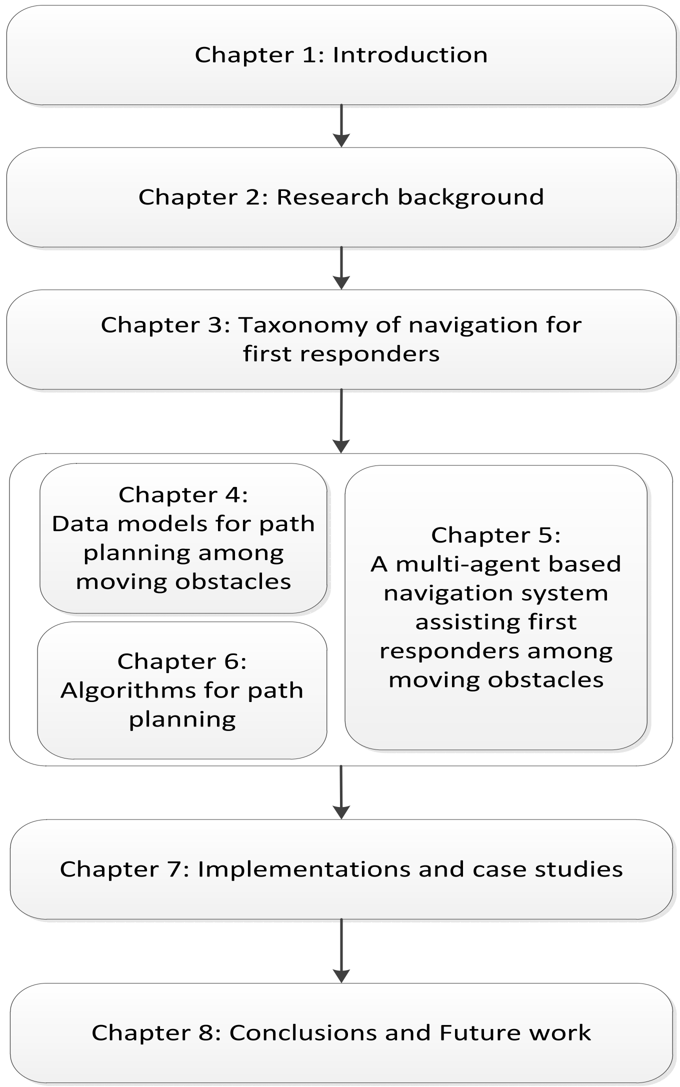



Figure 1.4: Outline of the thesis

# 2 RESEARCH BACKGROUND

In Chapter 1, the motivation, research questions, and research methodology are presented. This chapter provides an overview of existing works in relation to this research project. The chapter first gives a brief background on navigation among obstacles (see Section 2.1). Section 2.2 describes the formalization of emergency processes in disaster response. In Section 2.3, we show the state of the art of hazard simulation systems which are capable of using dynamic data. As uncertainty is an important aspect, issues on uncertainties in hazard simulations are also discussed in this section. In Section 2.4, we review previous research on geo-database management systems. Section 2.5 discusses the agent technology and its applications on disaster management and GIScienece. Finalnly, in Section 2.6, we review various routing algorithms that are developed for different types of path planning problems. This chapter is partly based on the following own publications: Wang and Zlatanova (2013c,b); Wang et al. (2014, 2015).

## 2.1 Navigation in the presence of moving obstacles

Navigation has been thoroughly studied from varied theoretical perspectives and across multiple disciplines, such as robotics, geomatics and applied mathematics (Chabini and Lan, 2002a; Ge and Cui, 2002; Huang et al., 2007; Delling et al., 2009). Advances in positioning technologies, such as GPS, radio frequency identification (RFID), wireless local area network (WLAN), and ultrasound range sensors (Khoury and Kamat, 2009; Girard et al., 2011; Li and Becerik-Gerber, 2011; Verbree et al., 2013), also provide significantly rich solutions to navigation related issues, although their focus and applications differ considerably. In the past few years, there have been a number of large-scale disasters causing tremendous economic losses and millions of victims, e.g., the Indian Ocean tsunami in 2004, Wenchuan earthquake in China in 2008, Fukushima nuclear power plant accident in 2011. In dealing with these disasters, the responders need navigation services that are capable of guiding them to avoid obstacles. For example, as shown in Figure 2.1, in the case of fires, when the responders are on the way to the incident site to perform their tasks, the fires could affect some roads in the areas, which may make blocks and slow down the rescue operations. Since the emergency navigation plays a special role in the disaster response and is vital for saving people's lives during disasters, there is a great need for investigation of issues related to navigation during disasters.


Figure 2.1: Navigation for fire trucks during the fires

Despite the considerable amount of route guidance research that has been performed, very few research efforts have been devoted specifically to emergency navigation problems in the context of moving obstacles that dynamically affect the road network (Wang and Zlatanova, 2013c). Many commercial navigation systems (e.g. Tom-Tom, Mio, Garmin) have been designed and developed to provide personalized routing services, and some of them are even able to incorporate information about traffic congestions and suggest alternative routes. However, these systems do not take into account specific emergency response requirements, which result in poor performance in response to disasters. The navigation service provided by existing emergency support systems (Parker et al., 2008; Johnson, 2008) are capable of finding the shortest route to a certain location, taking the damages of the infrastructure into account, but lack consideration of dynamics of disasters, which brings serious limitations to application of these systems in the road network dynamically affected by disasters. Some studies have investigated the possibility of using crowdsourced data to make a crisis map including the blocked areas (Gunawan, 2013). Using these crowdsourced information, Nedkov and Zlatanova (2011) propose a method that is able to guide Google's Directions Service around obstacles (see

Figure 2.2). For similar purpose, Schmitz et al. (2008) present a web-based route service called OpenRouteService (http://openrouteservice.org) which can provide route planning services taking blocked areas or streets into account. However, they can only cope with static obstacles, and do not offer the routing functionality required to avoid moving obstacles.

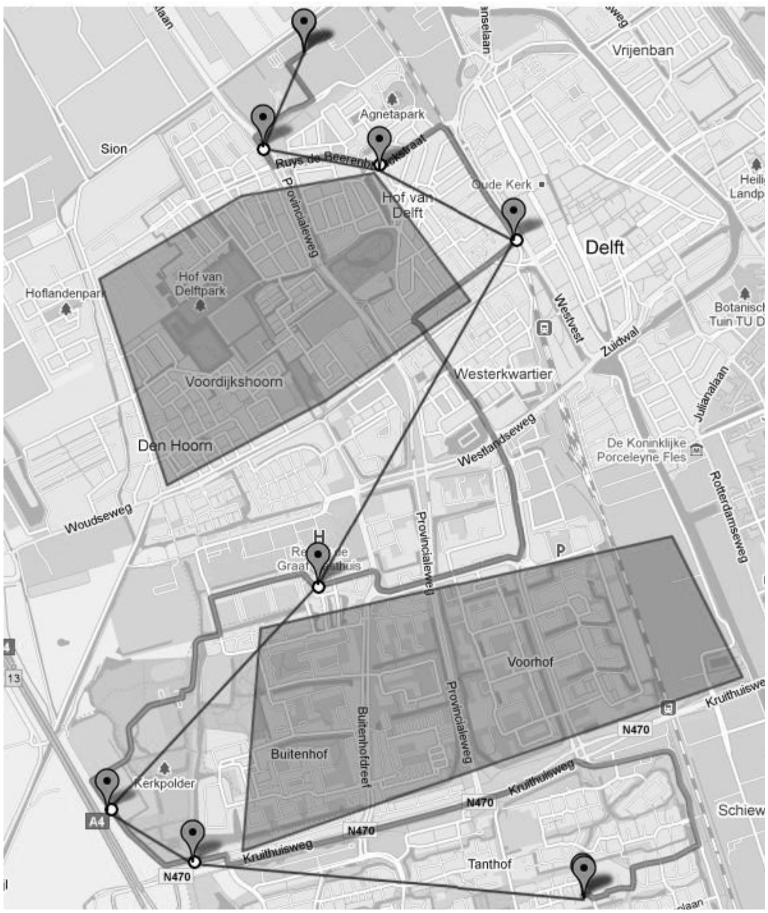



Figure 2.2: Calculated route avoiding defined obstacles (from Nedkov and Zlatanova (2011))

With the advance of disaster modeling and simulation technologies (Darema, 2004; Rodríguez et al., 2009; Moreno et al., 2014; Zelle et al., 2013; Lu et al., 2008), some researchers have tried to incorporate the disaster simulation to improve the routing process. Using hydrological model for flood prediction, Mioc et al. (2008) study the calculation of evacuation route under the flood disaster, considering vehicle types and the water depth on roads (see Figure 2.3). They also developed a prototype of web-based GIS application that allows individuals to request for evacuation routes (Mioc et al., 2012). Chitumalla et al. (2008) present an application that uses the forecast information of plumes in the near future in the routing and provides navigation services taking blocked areas or streets into account. Nevertheless, the considered obstacles are still stationary, which can not reflect the dynamics of physical phenomena (floods, plumes, fires, etc.) that cause disasters. This can make the planned path much longer than the shortest one. In some situations, the responders can pass through the threatened roads before they are affected instead of just avoiding them.


Figure 2.3: The evacuation route, taking into account water depths on roads (from Mioc et al. (2008))

On the other hand, most research on dynamic obstacles has been centered on robotics (Li et al., 2009; Gonzalez et al., 2012b; Yang et al., 2006; Belkhouche et al., 2007; Ni and Yang, 2011). The results from these studies could benefit the navigation of first responders in certain aspects. Phillips and Likhachev (2011b) introduce the concept of safe intervals to compress search space and to generate collision free paths in dynamic environments with moving obstacles. Masehian and Katebi (2007) present an online-based method to address the problem of multi-robot pursuing a moving target amidst both dynamic and static obstacles. The proposed method first generates a set of collision-free paths and divides possible directions into several parts, from which a near-optimal path to the target is selected for the robot to follow, as shown in Figure 2.4. For dynamic and uncertain environments, Sonti et al. (2013) develop a grid-based path planning algorithm that can use the stochastic model of dynamic obstacles as well as the onboard sensor measurements. However, most of the previous research in robotics focuses on the routing within the free space, which does not consider constrained movement on a transportation network.

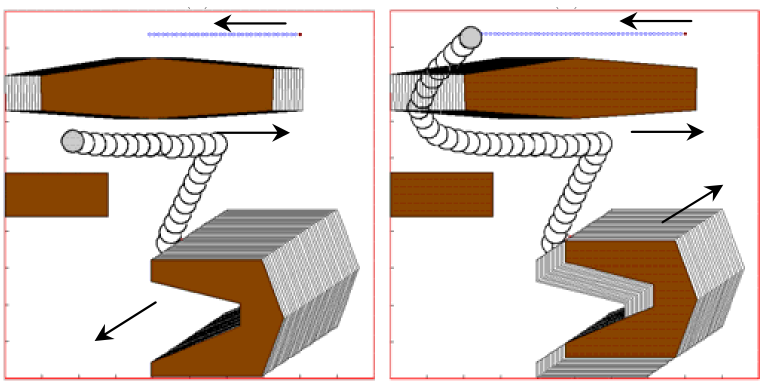


Figure 2.4: Collision-free trajectories of robot that pursuits a moving target in an environment with one static and two moving obstacles (from Masehian and Katebi (2007))

## 2.2 Formalization of the processes in emergency response

Emergency response procedures are designed for first responders to take appropriate actions to address incidents. A conceptualization of these emergency response procedures can help make a better understanding of the role of actors and their activities involved in the disaster response. Xu et al. (2008) discuss both the advantages and disadvantages of using different modeling languages in modeling disaster management processes. Zlatanova (2010) formally model the emergency processes and tasks in the Netherlands, using the Unified Modeling Language (UML). More importantly, this conceptualization can also serve as a guideline to the development of crisis management systems. Using the conceptual components in disaster management, geo-data models can be built within the systems, supporting data sharing among emergency actors (Aydinoglu et al., 2009). Special rules can also be defined to allow the systems to deliver relevant information that are required for emergency response processes (Fan and Zlatanova, 2011). For designing and developing an emergency navigation system, it is also necessary to take into account the emergency response processes that have been defined. This would make the navigation system able to extract the information essential for route planning and to generate routes customized based on the tasks of responders.

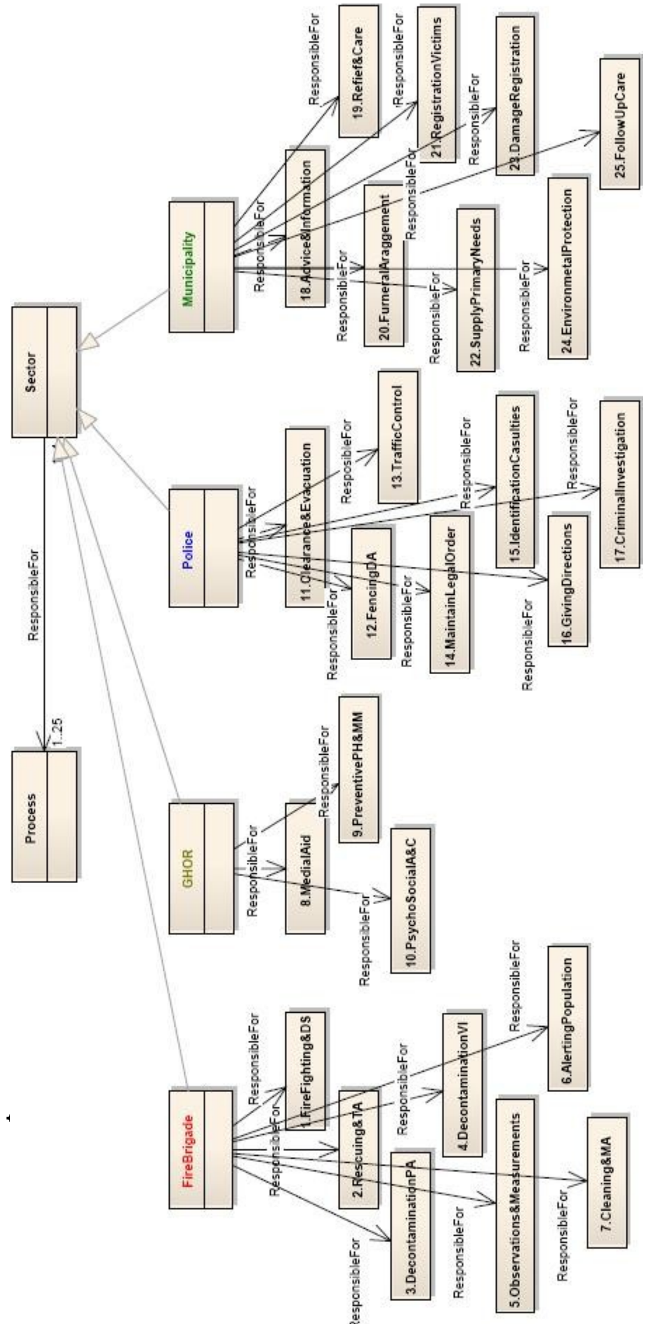



Figure 2.5: *Processes* defined for emergency response *sectors* in the Netherlands (from Zlatanova (2010))

As mentioned in Chapter 1, in the Netherlands, the emergency response procedures have been defined within the Dutch law (Dilo and Zlatanova, 2008). 5 GRIP (Coordinated Regional Incident Suppression Procedure) levels, i.e., GRIP 1-5, have been defined to guide the actions and tasks that should be executed during a disaster response (Van Borkulo et al., 2005). According to the GRIP level, different *Processes*, which consist of a set of *tasks*, are activated to manage the disasters. Figure 2.5 shows the 25 types of *processes* defined for the four *sectors*: police, fire brigade, municipality, and medical care. Each *sector* is responsible for a group of *processes*. *Actors* (individuals or teams) within the *sector* play different roles and perform operational *tasks*. Because fire brigade is one of the primary responders in the Netherlands, we take fire fighting as an example to illustrate the workflow of the fire brigade. In the case of large fires, the *fire fighting process* will be activated and works as follows. The Call Center first registers the incident after receiving the emergency call, and then informs the responsive fire brigade units to fight fires and to do measurement and observations. The officer on duty (OfficeDuty) and the fire brigade trucks leader (FBleader) move to the location of the fire incident. On the way to the fire they examine the needed information, such as vulnerable objects in the affected area, the number of injuries, and the locations of fire hydrants, etc., and request the fastest or safest route to the destination. If the severity of fire incidents rises to the level GRIP 2, a special Regional Operational Team (ROT) is formed to lead the actors involved in the fire fighting process and to coordinate their actions. Both OfficeDuty and FBleader have to report to the ROT.

## 2.3 DDDAS and hazard simulations

Hazard simulations, which provide the forecasted information of disaster situations, are essential in assisting emergency managers in making rescue plans. However, traditional simulation models based on rigid input parameters, which are largely decoupled with real systems and make little usage of real-time data, fail to reflect the real behaviors, and thus can not satisfy the accuracy requirements for disaster response planning. This section shows a simulation system paradigm that aims at using dynamically assimilated data to improve the modeling and prediction capabilities of applications, Dynamic Data Driven Application Systems (DDDAS). This paradigm will be briefly introduced in Section 2.3.1. After that, we give an overview of prior work on the application of the DDDAS approach in modeling hazards in Section 2.3.2. Finally, Section 2.3.3 discusses the issue of uncertainties that are involved in hazard simulations.

### 2.3.1 DDDAS

The concept of Dynamic Data Driven Application Systems (DDDAS) was proposed by Darema (2004) to improve the prediction results from applications. The idea that underlies DDDAS concept is to couple simulation models with the measurement process by using dynamic data in order to achieve more accurate simulation results. This approach seeks to continuously adjust the systems, using mathematical and statistical algorithms (e.g., Kalman filter, particle filter) to assimilate sensor data, and to conversely control the measurement process, providing the systems with better quality inputs. A number of research works have been carried out to address challenges of creating DDDAS capabilities. These include developments in a wide range of science and engineering disciplines, such as sensor networks (Jiang and Parashar, 2009; Bein et al., 2013), applications algorithms (Lucor et al., 2004; Hu, 2011), and system software (Douglas et al., 2006; Allen, 2007).

### 2.3.2 DDDAS-based hazard simulations

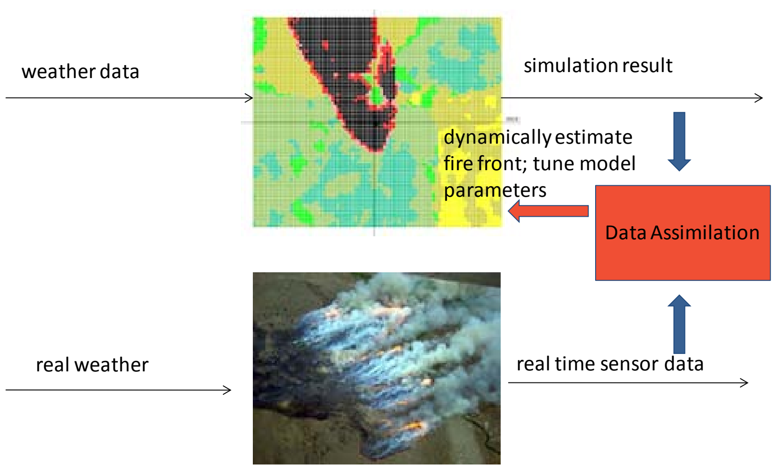



Figure 2.6: Dynamic Data Driven Simulation for Wildfire Spread Prediction (from Hu (2011))

Using the DDDAS concept, many hazard models have emerged in the past few years (Moreno et al., 2011; Rodríguez et al., 2009; Hu, 2011; Zelle et al., 2013). For example, Trafalis et al. (2004) develop a hybrid forecast system that can be corrected with dynamic data injection and steering. The use of continuous data streaming leads to the improved ability of the system in discriminating tornadic from nontornadic events. Zelle et al. (2013) present an integrated system for smoke plume and gas cloud forecasts, combining a weather model, a smoke plume model, and a crisis management system. Moreno et al. (2011) propose a real-time fire simulation algorithm that can be integrated into interactive virtual simulations where fire fighters and managers can train their skills. Hu (2011) presents a dynamic data driven simulation system developed for estimating the wildfire front (see Figure 2.6). It uses Sequential Monte Carlo (SMC) methods to incorporate sensor data into the fire simulation system, which allows the fire model to be dynamically adjusted to make better predictions of wildfire spread.

Driven by real-time data collected from the field, these hazard simulations are capable of providing reliable predicted information about disaster changes, and hence are valuable tools that underlie the solutions for many problems that arise in rescue planning. With these simulation models, emergency workers can perform more accurate spatial analysis, including assessing the potential impact of hazards, identifying dangerous areas that should be evacuated, and determining inaccessible or impeded roads, etc. As a result, more effective plans can be generated to curb damages and protect lives.

### *2.3.3 Uncertainties in hazard simulations*

In real disasters, it is usually difficult to get very accurate predicted information of hazards from hazard simulations. This is because that some model errors exist in the source spatial data and the real-time data collected from the field may have inherent uncertainties. For example, in prediction of forest fires, although real-time data can be obtained through communication and sensor network to drive the fire simulation model, a variety of factors would make prediction of fire-front difficult. Those factors could either be randomness in weather conditions, such as, winds, precipitation, and humidity, or errors in the models, like terrain model and land use model. To address issues regarding these uncertainties, many research efforts have been devoted to this direction. In this dissertation, we take into consideration two of the core components in the uncertainty research associated with our work: uncertainty modeling and error propagation.

Modeling uncertainty in geographic information is the first step in dealing with uncertainties in hazard simulations. Different types of uncertainties can be involved in the representation of spatial objects, such as position uncertainty, attribute uncertainty, temporal uncertainty, and so on (Shi, 2008). To model and quantify these uncertainties in spatial data, many researchers have used different techniques, such as spatial statistics, fuzzy theory, probabilistic model, and along with their efforts, a large collection of conceptual models and data models have been proposed for representing and describing uncertainties associated with spatial objects. Krüger and Lakes (2015) address the issues on

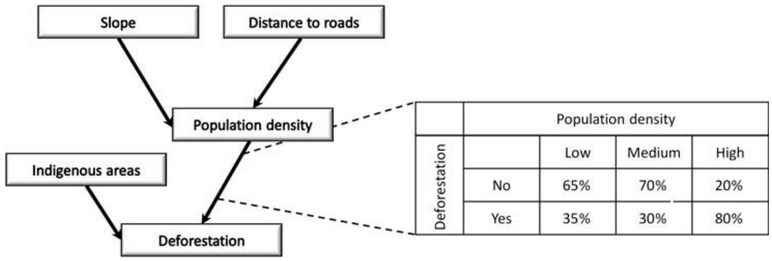



Figure 2.7: Example of a simple BBN, illustrating the dependency of "deforestation" on its associated variables

uncertainty in land-change modeling, using Bayesian belief networks (BBNs) (see Figure 2.7). Cheng (1999) investigates the modeling of fuzzy objects, addressing issues related to identification, detection, and representation of fuzzy objects and their dynamic changes. Similarly, using fuzzy set theory, Dilo (2006) defines different types of vague objects (vague points, vague lines, and vague regions), and a set of operators which allow reasoning with these vague objects. These works offer a foundation for understanding and assessing the impact of variations and errors of input data on the resulting output data from simulations.

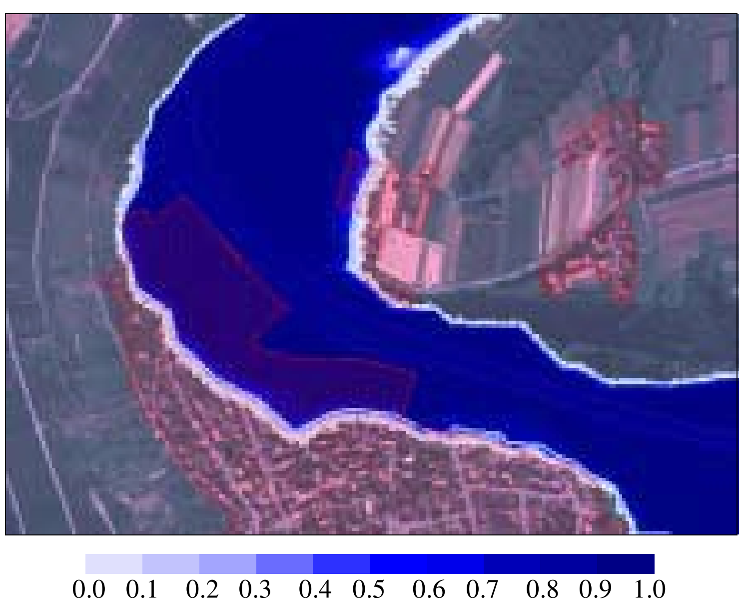



Figure 2.8: Representation of the probabilities of the flood region (from Glemser and Klein (2000))

Because hazard simulations make predictions based on the input data, the uncertainties in the original data source can have influence on the results obtained from simulation models. For decision making under uncertainties, there is a need for a way to assess and quantify the uncertainties that are propagated from source data. The effect of error propagation has also been investigated by GIS researchers. Many approaches have been applied for assessing and estimating the impact of the uncertain input data, including analytic method, Taylor series method, sensitivity analysis, Monte Carlo simulation, Bayesian Belief Network, etc. Glemser and Klein (2000) use Monte Carlos method to spatially quantify the probabilities of the flood region (see Figure 2.8). Shi et al. (2004) investigate the use of two methods (i.e., an analytical and simulation method) to model error propagation in spatial analysis of vector data. Ayre and Landis (2012) present a Bayesian network model for ecological risk assessment. The proposed Bayesian network structure allows it to incorporate different types of information, such as predictions, expert judgment, and uncertainties in stochastic ecological systems, into the model, which can reduce the uncertainties in the model. These approaches provide a promising way for studying uncertainty in hazard simulations. What's more, taking the resulting uncertainties into account, decision makers would draw more reliable conclusions from further analysis of results of hazard simulations.

## 2.4 Geo-DBMS

The previous section has shown the potential of hazard models to be a valuable addition in the disaster response. As the sensor measurements change over time, the predictions provided by the hazard models using the sensor information may also change rapidly, resulting in vast amounts of dynamic spatial data that needs to be structured. The Geo-DBMS, which has mechanisms that enable fast update and access to geographic information, is a suitable tool for management and sharing of large spatial data sets related to disaster management. This section first gives a short introduction of geo-DBMS and reviews previous work on applications of geo-DBMSs in the field of disaster management (Section 2.4.1). Then we provide an overview of the state-of-the-art research on management of moving objects, which provide support for managing dynamic data produced during the disaster response (Section 2.4.2).

### *2.4.1 Geo-DBMS and its application in disaster management*

Geo-database management systems (geo-DBMSs) are developed for storage, manipulation, and retrieval of geo-data from databases. The geo-DBMSs provide fast spatial access methods that can accelerate the geo-data retrieval

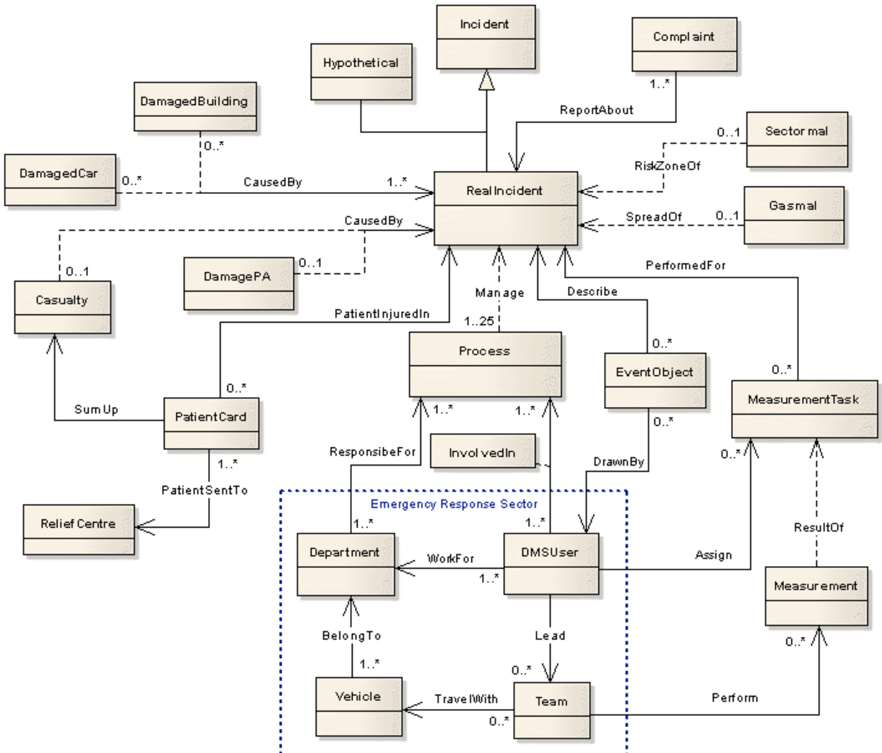



Figure 2.9: GDI4DM data model for emergency response in NL (from Dilo and Zlatanova (2011))

(Breunig and Zlatanova, 2011). More importantly, geo-DBMSs implement the 2D geometric and topological models that make them capable of handling different types of 2D spatial objects (such as point, line, region) and their relationships (Meijers et al., 2005). During the past several years, more attention is being paid to 3D-GIS (Stoter and Zlatanova, 2003). As a result, many 3D concepts and models have been proposed (van Oosterom et al., 1994; Zlatanova, 2000), and some of them have been implemented in geo-DBMSs for maintenance and query of 3D spatial objects (Stoter and van Oosterom, 2002; Penninga and van Oosterom, 2008). In addition to 2D and 3D models, geo-DBMSs also provide operations and functions on geo-data to define functionality for spatial analysis. With these models and functionalities, geo-DBMSs have been applied in many diverse domains, such as cadastre and land management (van Oosterom and Lemmen, 2002).

An important application of geo-DBMS is management of geo-information for disaster management (DM). Because many problems that arise in disas-

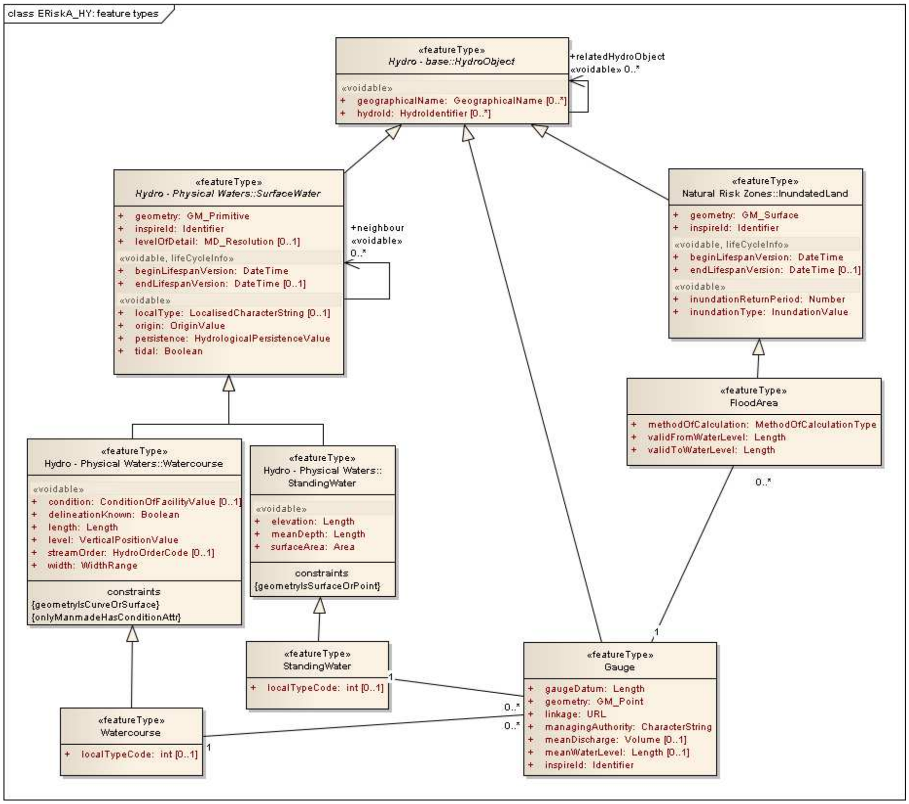



Figure 2.10: ERiskA data Model (from Zlatanova et al. (2010))

ter response are spatially relevant, geo-information is required in dealing with these problems (van Oosterom, 2009; Zlatanova and Holweg, 2004). Geo-DBMSs have been used in a multitude of decision support systems for disaster management to store the geo-information related to emergency operations (Kwan and Lee, 2005; Zlatanova and Baharin, 2008; Mioc et al., 2008). To facilitate organization and structuring of this information, a large group of data models have been proposed for different types of disasters, such as the Geographical Data Infrastructure for Disaster Management (GDI4DM) data model (Figure 2.9), the European Risk Atlas (ERiskA) data model (Figure 2.10), the Department of Homeland Security (DHS) geospatial model (Figure 2.11), and Border Security data model (Figure 2.12). And some of them have been implemented and tested in geo-DBMSs (Dilo and Zlatanova, 2011, 2008). Besides, some data models have also been developed for navigation in disasters. Zlatanova and Baharin (2008) design an emergency response network model to support representation of information regarding event, user, and spatial network. Kwan and Lee (2005) describe a 3D network data model that can integrate multi-level structures of buildings with the road network system. Base on this model, they built a GIS-based intelligent emergency response system that can assist routing for rescuers. However, the above data models are not capable of dealing with spatio-temporal information of road networks affected by hazards, and thus can not support navigation for responders in the presence of moving obstacles.

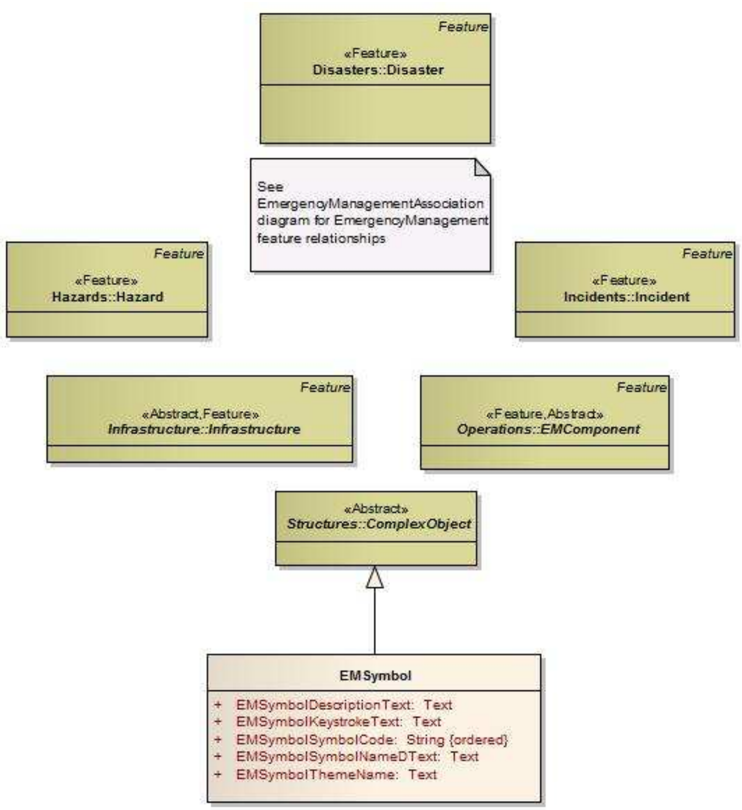



Figure 2.11: Department of Homeland Security Geospatial (DHS) data model (from Zlatanova et al. (2010))

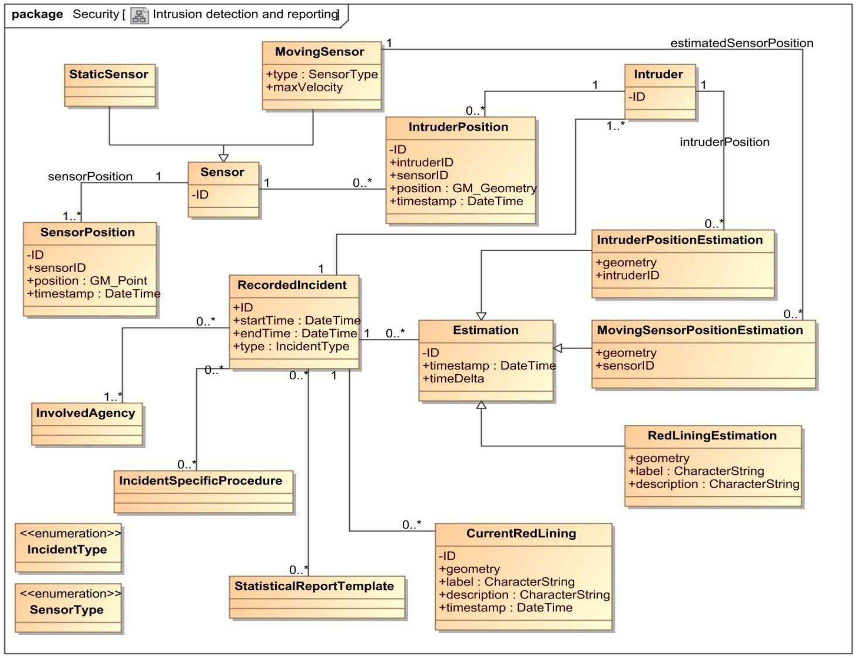



Figure 2.12: Border Security data model (from Zlatanova et al. (2010))

### 2.4.2 *Management of moving objects in geo-DBMS*

Routing during disasters is a complex process that involves many actors and hazards, which could produce huge amounts of dynamic data of moving objects. Typical examples of this dynamic information are: the prediction of hazards and the location of responders (Dilo and Zlatanova, 2011). Predictions of hazards are generated by the hazard models driven by real-time information (e.g, humidity, wind speed), and the location of responders is collected from the field by geographic positioning technologies. Both are useful for navigation avoiding moving obstacles. For instance, disaster managers need to keep track of location of responders on the field for monitoring purpose, and more benefits can be achieved from analyzing the moving object information for navigation purpose (e.g., derive the predicted speed of vehicles and include it in the routing). The frequent changes of such dynamic information demand an efficient management of moving objects in DBMS (Zlatanova and Baharin, 2008).

Many researchers have been working on managing moving objects, and numerous data management techniques have been developed to facilitate the collection, organization, and storage of dynamic data of moving objects

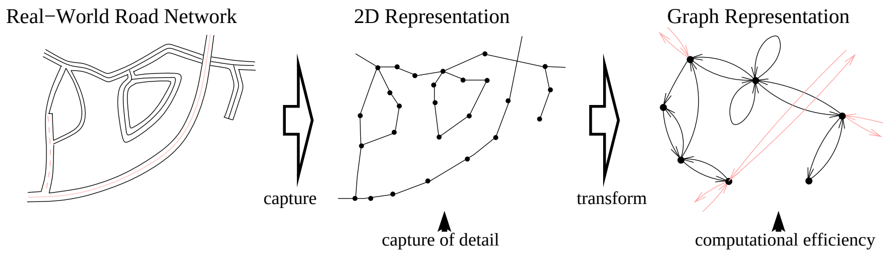



Figure 2.13: Two representations of road networks (from Speičvcys et al. (2003))

(Wolfson et al., 1998; Meratnia, 2005; Güting et al., 2006; Speičvcys et al., 2003; Šaltenis et al., 2000). Meratnia (2005) addresses a couple of issues in modeling, compressing, analyzing the data of moving objects. Sistla et al. (1997) propose a Moving Objects Spatio-Temporal (MOST) model which is capable of tracking not only the current, but also the near future positions of moving objects. Considering that the movement of users is often constrained by road networks, Speičvcys et al. (2003) present a computational data structure that models network constrained moving objects, using two representation (i.e., two-dimensional and graph) of roads networks and objects (see Figure 2.13). Besides, the indexing of moving objects has also been well studied (Šaltenis et al., 2000). De Almeida and Güting (2005) propose an R*-tree based indexing technique that supports the efficient querying of the current and projected future positions of moving objects. In real disasters, large volume of information of the dynamic objects (e.g., the locations of the rescue unit, plume movement, and changes in the water level) would be generated from sensors and hazard simulations, and need to be stored in the database. The above studies provide a rich set of methods for fast searching and retrieving the needed piece of information from the database, supporting management of huge and rapidly growing datasets produced during disasters.

## 2.5 Agent technology

The agent technology was introduced by Wooldridge and Jennings (1995) and represents an approach for development of software entities that automates specific tasks. While a single agent, who has a set of properties, such as autonomy, rationality, reactivity, proactiveness, can handle relatively simple problems, a multi-agent system (MAS), which consists of a network of agents, is able to cooperate with each other to address more complex problems. The agent technology has been applied in a very varied of fields, for example, crisis management (Schoenharl and Madey, 2011; Schurr et al., 2005), supply chain

(Vokřínek et al., 2010; Teodorović and Pavković, 1996), and information management (de Bruijn and Wijngaards, 2013; Genc et al., 2013a). The following sections provide a literature review on previous studies on two important applications related to this research: disaster management and GIS.

### *2.5.1 Agent technology and disaster management*

Agent technology has been used in all phases of disaster management (DM), including preparation, response, recovery, and long-term mitigation. Fiedrich and Burghardt (2007) distinguish two types of agent systems designed for disaster management:

AGENT-BASED SIMULATION SYSTEMS which are systems for simulating people's activities, behaviors, and their interactions during or after disaster events

AGENT-BASED DECISION SUPPORT SYSTEMS which are used to provide support for emergency managers in the decision-making processes at various levels

In the first type of agent systems, agents are designed based on a set of behavioral rules, and are able to interact with other agents to model human behaviors and interactions in complex disaster scenarios. These simulation systems allow emergency managers to better understand the dynamics of disaster situations, and to evaluate possible plans and procedures for dealing with disasters. Several related research projects have been conducted, including Robocup Rescue (Kitano et al., 1999), DEFACTO (Schurr et al., 2005), and DrillSim (Balasubramanian et al., 2006). Besides, a significant number of research studies have investigated the use of agent technology in DM. Zarboutis and Marmaras (2007) present a methodological framework that uses agent system to model the triple interaction between personnel, hazard, and evacuees in a metro system, and to provide recommendations in designing formative evacuation plans. Filippoupolitis et al. (2011) use a multi-agent simulation platform to evaluate performance of the proposed navigation systems in indoor evacuation. Each actor (e.g. civilian) involved in the scenarios is represented by an agent who takes actions based on behaviour models (e.g. health and movement models of civilians). Ehlen et al. (2014) develop an agent-based chemical supply chain model to study the impact of a hurricane disruption. Their agent model is integrated with other models, such as market model, transportation model, to capture the essential dynamics of the supply chain component.

The second type of agent systems consists of agents that are equipped with a variety of functionalities for crisis decision making, such as information processing and analysis, use of optimization algorithms, and integration of collaboration mechanisms. Many efforts in DM have been directed toward this research area. Sheremetov et al. (2004) build a contingency management system based on an agent-based intelligent infrastructure. Distributed coalition formation model is implemented in the agent model to solve the logistics planning for personnel evacuation. Ibri et al. (2012) propose a decentralized distributed solution approach using multi-agent systems (MAS), with the aim to solve emergency vehicle dispatching and covering problems jointly. Buford et al. (2006) implements a multi-agent architecture which is composed of Belief-Desire-Intention (BDI) agents with capability of situation awareness, supporting situation management in large-scale disasters.

As mentioned earlier, the changes of obstacles could generate large amounts of data, which need to be processed and analyzed. Therefore, it would be interesting to explore the use of agent technology in performing automatic processing and analysis of these data, assisting responders in decision making during disasters.

### *2.5.2 Agent technology and GIS*

Recently, there is an increasing interest in the research of application of agent technology in GIS domain. Many researchers have been focused on developing various types of agents for GIScience applications, for example, map generalisation (Lamy et al., 1999; Duchêne et al., 2012). To capture the diversity of agents emerged from their research works, Sengupta and Sieber (2007) propose two broad types of agents that predominate in GIScience: Artificial Life Geospatial Agents (ALGAs) and Software Geospatial Agents (SGAs). Using this categorization, we briefly review the literature on combination of agent technology and GIS.

#### *1. Artificial Life Geospatial Agents (ALGAs)*

The agents of this type can mimic the behavioral response of entities (e.g., people, animals, organizations), and simulate the interactions between them within social/spatial networks. This type of agent system provides a way to express dynamics related to spatial elements, which can not be offered by traditional GIS tools. With spatial models in building dynamic simulations, some researchers have developed ALGAs to simulate the behaviors of human in interaction to the spatial environment. Schoenharl et al. (2006) develop an agent-based simulation system that can be integrated with real-time sensor

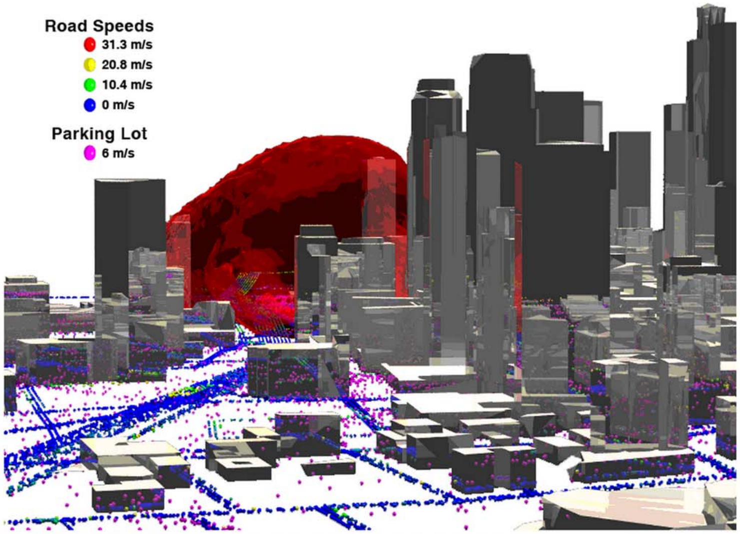



Figure 2.14: Snapshot of simulation model with buildings (gray polygons), the plume (in red), and agents as spheres color-coded by speed (from Epstein et al. (2011))

data, which allows it to predict possible outcomes based on the current status of crisis situations. Tang and Zhang (2008) present an agent-based simulation model to study evacuation process in the building environment. The GIS-based environmental analysis is used to configure the cognition of agents. Epstein et al. (2011) combine Agent-Based Modeling (ABM) with Computational Fluid Dynamics (CFD) to study human behaviors in responding to contaminant plumes in an urban environment, as illustrated in Figure 2.14.

*2. Software Geospatial Agents (SGAs)*

SGAs serve a wide range of purposes, including management of geographical information and making spatial decisions. They are software agents that are incorporated with knowledge about spatial data modelling and its related issues, and are capable of automatically transforming, manipulating, and interpreting spatial data on behalf of users. Because many decisions and activities involved in the disaster management require large amount of spatial data, SGAs are powerful tools for tackling spatial problems that arise in response to disasters. Genc et al. (2013b) propose an agent-based infrastructure to assure secure flows of information, which includes geo- and contextual information produced in disasters. The proposed infrastructure uses software agents in exchange of information and automated data processing. El-Korany and El-Bahnasy (2008) present a multi-agent based crisis management system for fire fighting and suppression. The system contains a GIS agent which is responsible for providing information for different emergency services issues and delivering the map of fire areas. Nourjou et al. (2013) design a GIS-based assistant software agent that can assists the incident commander in various tasks, including strategic planning, centralized scheduling, and state-space search.

To be able to handle the spatial information related to the moving obstacles, specific types of software agents would be needed. Special GIS functionalities should also be designed and developed within the agents for spatial data processing and analysis. An important function of them is the path planning, which will be described in the next section.

## 2.6 Path planning algorithms

Quick and reliable routing algorithms are the core of emergency navigation systems. When incidents occur, the structure of road networks and their availabilities vary with the development of physical phenomena (e.g., floods, plumes, fires) that are caused by disasters, which makes the path finding for the relief vehicles in this dynamic environment quite difficult. Basically, the navigation for first responders covers three types of path panning problems: 1). one-to-one (one object has to be routed to one destination), 2). one-to-many (one object has to be routed to many destinations), and 3). many-to-many (many objects have to be routed to many destinations). For solving each type of routing problems, there have been a tremendous amount of research studies carried out, and a great number of routing algorithms have been developed for different navigation purposes.

### ONE-TO-ONE

This is a typical routing problem that involves a pair of source and destination. Various algorithms have been designed for this type of problem for a wide range of navigation purposes, ranging from classic algorithms, such as Dijkstra's algorithm (Dijkstra (1959)), A* algorithm (Hart et al. (1968)), Bellman-Ford algorithm (Bellman (1958)), to heuristic techniques, such as particle swarm algorithms (PSO), neural network (NN). In this dissertation, we study the one-to-one path planning among moving obstacles, which is a fundamental component in the navigation system for first responders.

The path planning among moving obstacles (e.g. floods, plumes, fires) poses a series of new challenges for researchers in the navigation field. Although many algorithm have been proposed for finding shortest routes in dynamic road networks, but there have been very few works that investigate routing in dynamic road networks affected by moving obstacles. Along with the advancement of intelligent transportation systems (ITS), people have developed a variety of routing algorithms to take into account traffic conditions and time-dependent travel times (Nannicini et al., 2012; Chabini and Lan, 2002b; George et al., 2007). These algorithms are capable of finding fast route in the dynamic network affected by traffic flows, and some of them even can deal with certain obstacles, such as, traffic Jams and accidents. However, the algorithms developed in ITS do not consider the profile of users and the nature of hazards, which makes them not suitable for application in real disasters. In the disaster response, the information of responders plays a special role in the navigation among obstacles. For example, in the case of smoke plumes, responders can either pass through or have to avoid of toxic gases, depending on the available protective equipment and the amount of oxygen they have. Therefore, the profile of responders should be incorporated in the route determination during disasters.

Because the status of components of the road network (i.e., roads and junctions) varies with changes in moving obstacles, it is also necessary to take into consideration the moving obstacles in the path finding process. In the field of disaster management, Visser (2009) develops an obstacle avoiding routing algorithm that incorporates dynamic blocks caused by moving obstacles, but it can only deal with roads that have one or two blocks, which limits its application in complex situations. Liu et al. (2006) study the calculation of evacuation route under the flood disaster, considering vehicle types and the effect of water depth on walking speed respectively. However, they only focus on the routing in the case of flooding, taking its specific characteristics into consideration, which limits their application to other types of disasters, e.g. plumes.

In robotics, a large collection of algorithms have been developed for navigating robots among moving obstacles. By adding time as a third dimension in search space, Van Den Berg and Overmars (2005) propose a two-level search method for robots to find a trajectory without any collision with moving obstacles. Inspired by the work of Van Den Berg and Overmars (2005), Phillips and Likhachev (2011b) use the concept of safe intervals to reduce the number of states in search space, and present a A* based planner that can incorporate waiting options in the search with safe intervals, as illustrated in Figure 2.15. For robots that have a limited battery power and are moving in dynamic environments, Phillips and Likhachev (2011a) develop an algorithm called Cost Function Dependent Action A* (CFDA-A*) that can remove cost function from the state variable while maintaining sub-optimality. Rufli et al. (2009) develop an approach that combines a pre-computation technique with an anytime search algorithm to search globally optimal route in constrained spaces

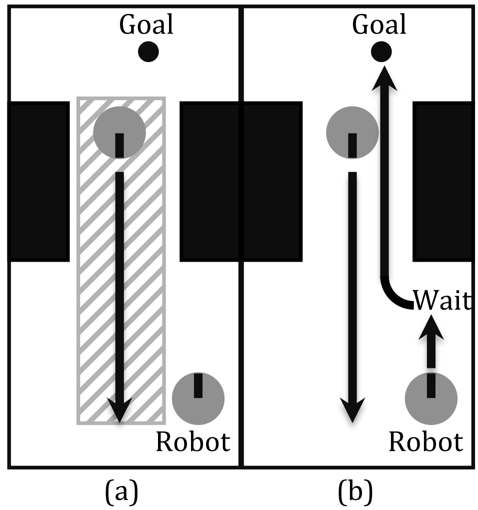



Figure 2.15: (a) Path planning considering the dynamic obstacle as a static one, which results in no solution; (b) Path planning with time, which allows the robot to wait to avoid the obstacle (from Phillips and Likhachev (2011b))

with dynamic obstacles. Although concentrating on the routing in free space, the above works could be beneficial to the research on navigation for first responders in various aspects, and offer potentially valuable resources for developments of algorithms for path planning in road networks affected by moving obstacles.

#### ONE-TO-MANY

Basically, this can be seen as a Travel Salesmen Problem (TSP). The objective of this problem is to find an optimal or near-optimal path to a set of destinations, the visiting order of which is not pre-determined. TSP is an NP-hard combinatorial problem, and is very difficult to be solved optimally in a short computational time, especially when there is a large amount of destinations. Several techniques have been used to assist in finding a cost-effective solution, ranging from techniques based on mathematical programming, such as cutting plane methods, and branch-and-bound, to techniques that use heuristics or swarm intelligence, such as incremental insertion mehtod, neural network (Saadatmand-Tarzjan et al., 2007), genetic algorithms (Holand, 1992), ant colony algorithm (Dorigo and Gambardella, 1997; Chen and Chien, 2011). The complexity of calculation is increased when the environment is affected by moving obstacles, which requires consideration of temporal aspect in planning the TSP tour.

MANY-TO-MANY

The problem of many-to-many path planning can be formulated as the Multiple Travel Salesmen Problem (MTSP), which is also computationally hard problem. This type of problems is characterized by involvement of a number of objects (e.g., vehicles, robots) and a set of destinations, which requires algorithms to be capable of routing these objects to their destinations in an efficient way. For addressing this problem and its variants, many algorithms and mechanisms have been proposed in various disciplines, especially logistics and robotics.

In the field of logistics, researchers extend the MTSP problem to the vehicle routing problems (VRPs) where a fleet of vehicles with limited capacity have to deliver or pickup goods from one or more depots to a set of cities or customers. A large number of techniques, such as fuzzy logic approach, evolutionary algorithms, have been used and developed to handle some additional constraints and to achieve efficient allocation of vehicles in a wide variety of VRPs (Vokřínek et al., 2010; Teodorović and Pavković, 1996; Prins, 2004).

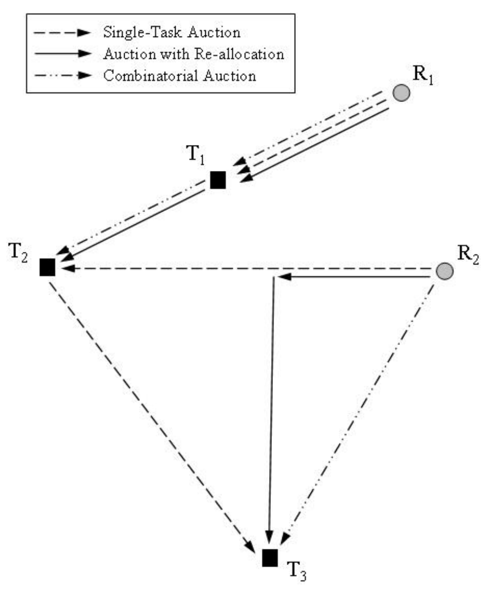



Figure 2.16: Task allocation using a single-task auction, a combinatorial auction, and the auction based on re-allocation of tasks (from Nanjanath and Gini (2006))

In robotics, the path planning for robots is regarded as the task allocation problem where a number of tasks have to be assigned to a team and have to be accomplished by the members. The considered tasks are simply locations that should be visited by the robots. Auction-based approaches have been widely used for addressing the task problems for robots (Dias et al., 2006).

Koenig et al. (2006) presents an approach based on Sequential Single-Item (SSI) auctions for multi-robot routing. All available tasks are allocated to robots through a sequence of auctions. Zheng et al. (2006) investigate the use of techniques (i.e., larger lookahead and rollouts) in SSI auctions to improve the evaluation of partial target assignments. Due to situations changes and robot failures, the tasks need to be re-allocated, which can be considered as a dynamic task allocation problem. To address the issues caused by unexpected events, Nanjanath and Gini (2006) present an auction-based method which allows rebidding on the un-accomplished tasks. Figure 2.16 gives an example comparing their method with other allocation mechanisms. Schoenig and Pagnucco (2011) evaluates the effects of different auction schemes on the optimality of the dynamic task allocation results.

All of the above works, which are developed for the three types of navigation problems, provide a rich repository of knowledge for studying the path planning problems that involve one or multiple responders as well as moving obstacles.

# 3 TAXONOMY OF NAVIGATION FOR FIRST RESPONDERS

As the previous chapter shows challenges in the navigation field that arise during the disaster response, there is a need to classify and analyze various navigation cases to better support navigation for first responders. This chapter presents a taxonomy of navigation among obstacles, categorizing navigation cases on basis of type and multiplicity of responders, destinations, and obstacles. Firstly, we list two general requirements that are required for navigation during disasters (see Section 3.1). Then we introduce our taxonomy of navigation for first responders, and present the selected criteria for the taxonomy (see Section 3.2). To help us describe different navigation cases, we divide them into two broad categories according to the characteristic of obstacles: static obstacles and moving obstacles. In Section 3.3, we analyze the navigation cases with static obstacles. Section 3.4 gives discussions on the navigation cases with moving obstacles. We summarize our investigation on previous work related to these navigation cases in Section 3.5. Our investigation reveals some limitations in current research on navigation, and shows challenges that have not been explored yet. After that, we present our approach to the problem of navigating first responders among moving obstacles and describe the architecture of our proposed navigation system (see Section 3.6). Finally, we conclude this chapter in Section 3.7. This chapter is based on the publication Wang and Zlatanova (2013c).

## 3.1 General requirements of navigation for first responders

Although over the years researchers have put much effort into addressing all relevant aspects of navigation, navigation for first responders brings forwards requirements of a higher level, all of which are not met by existing navigation developments. Here we list two general requirements that should be taken into account.

*1. Considering the influence of hazards*

In a transportation network affected by disasters, road conditions could be changed drastically by many factors, such as fires, plumes, landslides and floods, which may cause one or more road segments to be unavailable or less usable during specific periods of time. For instance, in the context of a chemical plant explosion that results in many moving contaminant plumes, these moving plumes can be considered as obstacles with changing shapes and positions. When moving towards the incident site, emergency response units should not be guided right through the toxic plumes by their route planners.

*2. Coordinating multiple first responders*

As it was mentioned in Introduction (Chapter 1), the disaster response involves many collaborative activities among different agencies, which requires coordination of their routes and destinations. In many emergency situations, the first responders work in groups and cooperate with each other in performing emergency tasks. They need not only to obtain individual routes but also to take into consideration other units in the area. For example, in the case of emergency medical service, ambulances are distributed to different destinations to pick up and deliver patients, according to factors such as the situation of patients, the deployment of paramedics, availability of medical supplies in hospitals, etc. Therefore, there is a need for building an emergency navigation system that that can not only provide fast and safe routes but also efficiently allocate multiple responders to different locations.

## 3.2 Taxonomy of navigation with obstacles

This chapter presents a taxonomy of navigation cases and elaborate on the issues related to optimal navigation for mobile rescue units. This work represents the first step of an approach to support navigation for first responders. Our ultimate goal is to provide path-finding methods that can assist responders in navigation among static and moving obstacles.

In order to understand navigation cases in disasters, we need to categorize them and group them into different classes. More importantly, by introducing a comprehensive review of categories, we can gain a greater understanding of characteristics and differences of these cases and study them separately. This taxonomy also encourages the design of new techniques by taking advantage of achievements in relevant fields. In previous navigation research, Zlatanova and Baharin (2008) present a taxonomy of navigation, trying to structure this field into different categories. Nevertheless, this classification does not take into account the obstacles. By extending their work, we try explore the possible navigation scenarios that could exist in disasters

Instead of a strict classification (which is very difficult to provide), we offer some broad keywords and phrases that characterize some classes of these cases. We assume that: 1). We deal with moving objects (e.g. responders), but they start moving at a given time and from given positions 2). We have a variable number of obstacles, i.e. the routing should be able to deal with many obstacles. Following these assumptions, we have identified the following criteria and distinguish the cases in the form of a quadruple:

$$< X_1, X_2, X_3, X_4 >$$

where

(1) $X_1$ is the number of responders (One or Many)

(2) $X_2$ is the number of destinations (One or Many)

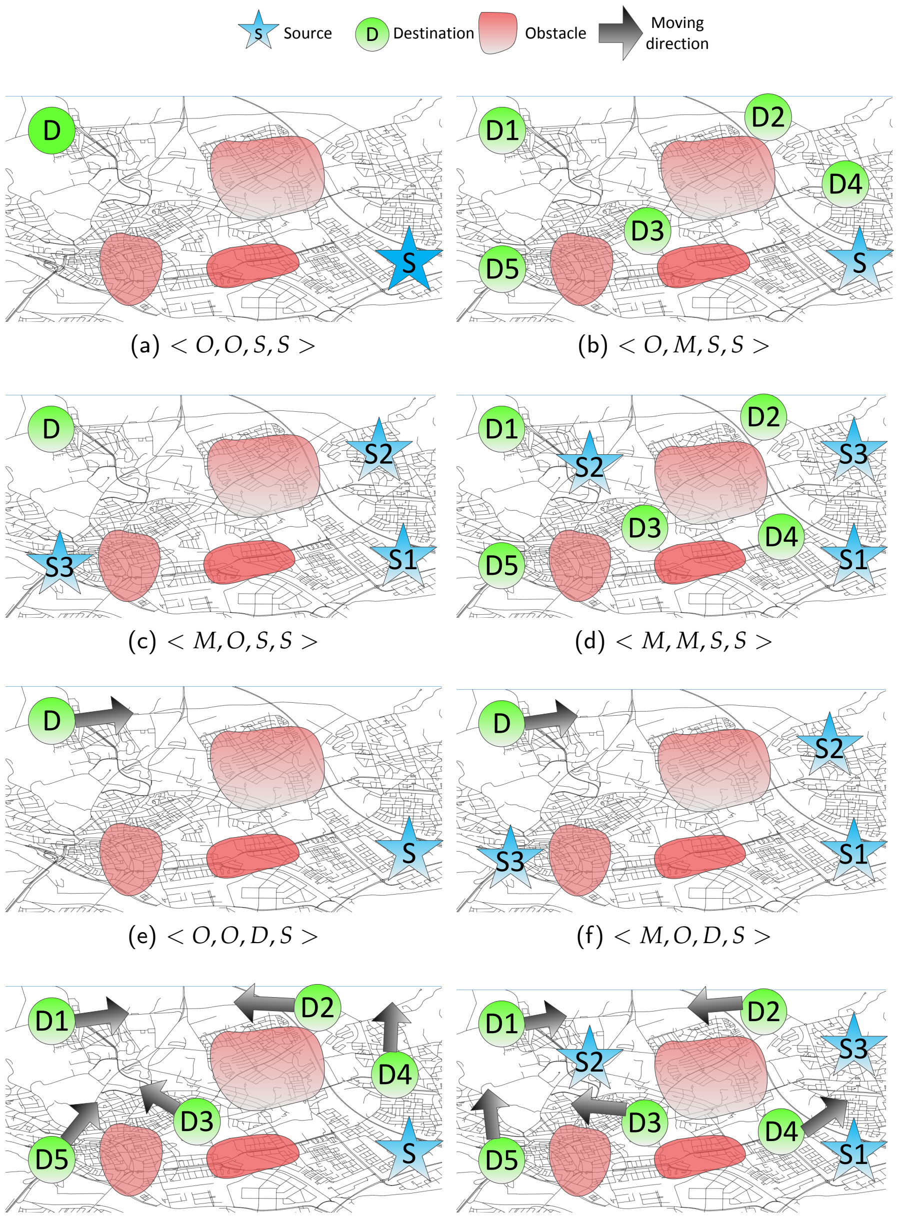
(3) $X_3$ is the type of destinations (Static or Dynamic)

(4) $X_4$ is the type of obstacles (Static or Moving)

For example, one case denoted by $< O, M, D, M >$ means one moving object has to be routed to many dynamic destinations, avoiding many moving obstacles. It should be noted that both "dynamic" and "moving" refer to the changes in geographic positions. We do not claim that this taxonomy is complete, since further refinements can be performed if adding more criteria, e.g. the obstacle can change its shape or not; the movement of the obstacle can be a priori known or not; the obstacle can have either distinct boundaries or fuzzy shape; people have to be routed to a destination point or a safe area. However, most current navigation cases fit into our taxonomy. In the following sections, We offer an overview of work that fits to each case and illustrate its potential application to emergency response.

## 3.3 Navigation cases with static obstacles

In this section, we mainly consider the navigation problems that involve static obstacles (see Figure 3.1), which are $< O, O, S, S >$, $< O, M, S, S >$, $< M, O, S, S >$, $< M, M, S, S >$, $< O, O, D, S >$, $< M, O, D, S >$, $< O, M, D, S >$, $< M, M, D, S >$.




(a) $< O, O, S, S >$ (b) $< O, M, S, S >$

(c) $< M, O, S, S >$ (d) $< M, M, S, S >$

(e) $< O, O, D, S >$ (f) $< M, O, D, S >$

(g) $< O, M, D, S >$ (h) $< M, M, D, S >$

Figure 3.1: Examples of navigation among static obstacles

$< O, O, S, S >$ *One moving object has to be routed to One Static destination, avoiding many Static obstacles*

This is a typical navigation example in rescue operations when disasters occur. A rescue team has to be routed through an affected area in which some segments are not passable, as shown in Figure 3.1a. In the outdoor environment, this navigation problem can be adapted by increasing the cost of affected roads to a very big value or excluding the affected roads, and solved by traditional routing algorithms. For the indoor environment, this case has also been well studied in computational geometry. The most basic form of the problem in a geometric context is: given a collection of obstacles, find a Euclidean shortest obstacle-avoiding path between two given points.

$< O, M, S, S >$ *One moving object has to be routed to Many Static destinations, avoiding many Static obstacles*

This situation may occur when a large-scale disaster affects different places and damages roads and bridges. For example, after a strong earthquake, a rescue unit is sent to deliver relief goods to several affected locations, avoiding untraversable roads (see Figure 3.1b). This problem can be addressed as a variant of the traveling salesman problem (TSP), the goal of which is to plan a trip with least cost in an environment with obstacles. The information of severely damaged transportation infrastructure is essential in the path planning for emergency response. Because in TSP the order of visiting destinations usually is not pre-determined, which makes this problem a NP-hard problem, an efficient route search algorithm is also needed to optimize the cost of the path visiting the given emergency locations.

$< M, O, S, S >$ *Many moving objects have to be routed to One Static destination, avoiding many Static obstacles*

This situation often takes place after the occurrence of disasters. A classical example is that several fire trucks have to be routed to a fire location through a road network damaged by earthquake or floods, as depicted in Figure 3.1c. This problem can be split into sub-problems by navigating moving objects separately, which can be addressed by approaches proposed for $< O, O, S, S >$. But in many circumstances, the task at the destination requires cooperation between responders. Therefore the navigation system should be able to assist emergency managers in selecting a certain number of responders that can arrive at the desired place at the same time. Moreover, as it is usually not possible to know in advance the location of obstacles, the system should also support sharing of information related to the road network to facilitate routing among obstacles.

$< M, M, S, S >$ *Many moving objects have to be routed to Many Static destinations, avoiding many Static obstacles*

This is the case that also often takes place when major disasters strike. Several rescue units are dispatched to transport a large amount of relief goods to different destinations, and each destination should be visited by at least one response unit, as exemplified in Figure 3.1d. In the field of logistic, substantial efforts have been made to provide the optimal set of routes for fleets of relief vehicles. Because disasters often cause traffic jams and accident, the difficulty of this problem is increased when these obstacles are taken into consideration. The sensor and communication technology for monitoring traffic conditions provides a promising method to overcome this difficulty.

$< O, O, D, S >$ *One moving object has to be navigated to One Dynamic destination, avoiding many Static obstacles*

This navigation case appears when one responder pursues one victim moving in a road network, parts of which are damaged and not accessible, as shown in Figure 3.1e. It also happens in the free space of a building in which corridors can be blocked by collapsed ceilings or floors. In many situations, not all of the information of obstacles is known prior to the departure of the vehicle. As more information about obstacles from sensors or communication network arrives, the system should be able to integrate this information into the routing. Besides, the interpolation technique would be needed for the system to analyze the trajectory of the target and to provide its predicted locations. This would help the responder to derive efficient strategies to intercept the moving target.

$< M, O, D, S >$ *Many moving objects have to be routed to One Dynamic destination, avoiding many Static obstacles*

Figure 3.1f depicts another situation that many first responders have to be routed through a transportation infrastructure ravaged by disasters to meet somewhere to exchange equipment or transfer the wounded. This problem can be seen as an extension of the well-known Pursuit-Evasion (PE) problem with obstacles, where the responders have to pursue a meeting point that changes with traffic conditions. Both the information of obstacles and vehicles (e.g., position, speed, departure time ) are needed to estimate the optimal or near-optimal meeting point. Another example is that some police cars are sent to stop a criminal trying to escape, avoiding traffic jams. At least one of the police cars should be able to capture the criminal. In this situation, the sharing of information of the moving target would facilitate the coordination between responders in pursuing the same target.

$< O, M, D, S >$ *One moving object has to be navigated to Many Dynamic destinations, avoiding many Static obstacles*

$< O, M, D, S >$ happens when a response unit is sent to rescue multiple victims that fleet using traffic facilities hit by natural disasters, as illustrated in Figure 3.1g. In this situation, the response unit can be considered as an interceptor and has to be navigated to pursue many targets whose positions change with the change of conditions. To be able to track victims, a sensor network that can connect different mobile devices, e.g., drones, cameras, phones, should be built. Because disasters often cause panic, which influences people's movements and actions, the use of models that can simulate the human behaviors during disasters would help develop more effective strategies for intercepting the targets.

$< M, M, D, S >$ *Many moving objects have to be routed to Many Dynamic destinations, avoiding many Static obstacles*

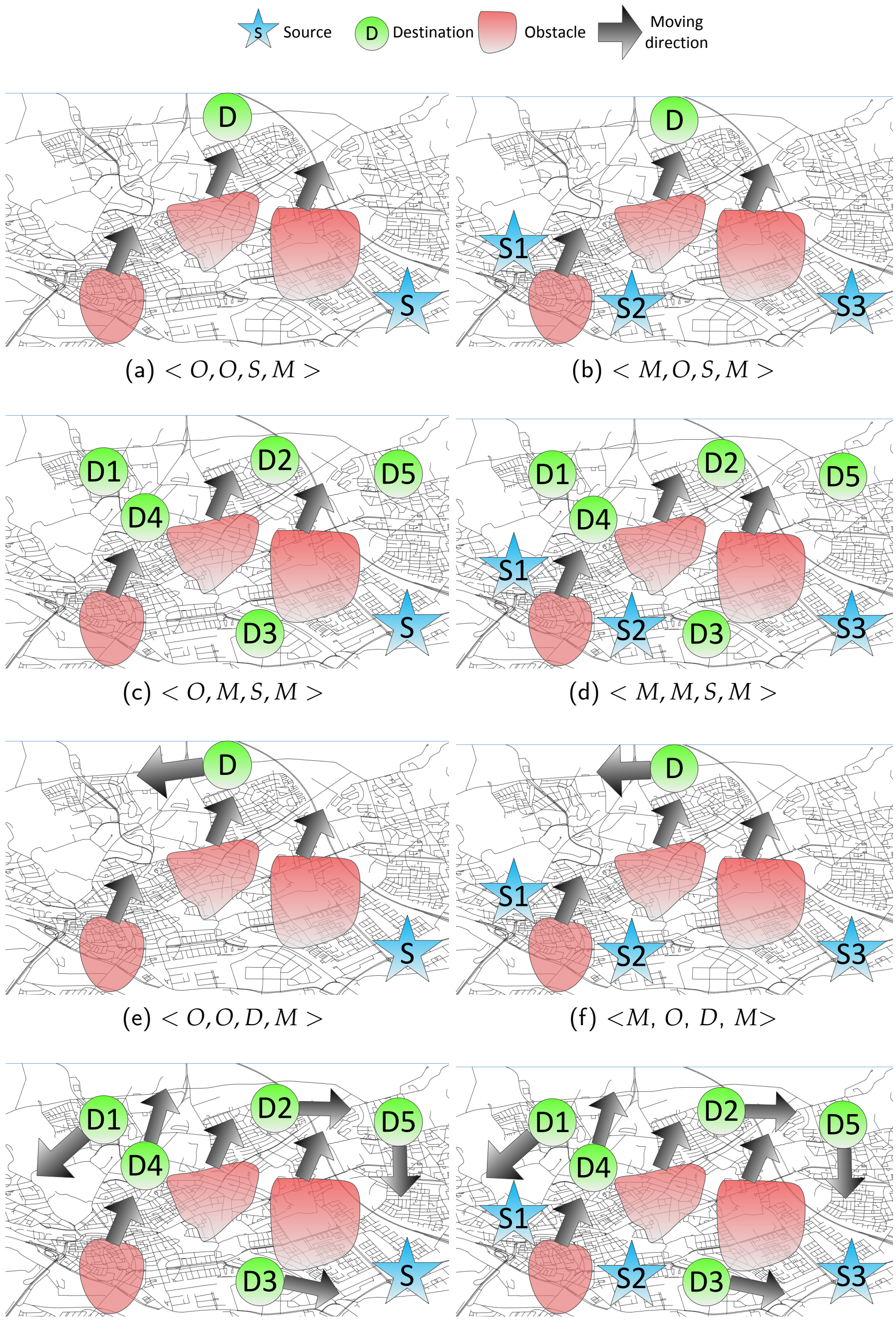
The situation in Figure 3.1h becomes more complex when several responders starting from different positions have to meet at a series of dynamic locations to perform their tasks, avoiding static obstacles. The path planning problem in this case is characterized by coordination among responders to reach the meeting points efficiently. One of possible solutions is to divide responders into different groups and to make them form coalitions in the pursuit of targets. These coalitions can be dynamically reformed according to new information on real situations during the execution of route plans.

## 3.4 Navigation cases with moving obstacles

Based on the literature reviewed, in this section we present and discuss navigation cases which involve moving obstacles, i.e., $< O, O, S, M >$, $< M, O, S, M >$, $< O, M, S, M >$, $< M, M, S, M >$, $< O, O, D, M >$, $<M, O, D, M>$, $< O, M, D, M >$, $< M, M, D, M >$, as shown in Figure 3.2.

$< O, O, S, M >$ *One moving object has to be routed to One Static destination, avoiding many Moving obstacles*

This may happen when one first responder has to go through an area affected by many moving obstacles (e.g. floods, fires) simultaneously (see Figure 3.2a).




(a) $< O, O, S, M >$ (b) $< M, O, S, M >$

(c) $< O, M, S, M >$ (d) $< M, M, S, M >$

(e) $< O, O, D, M >$ (f) $<M,\ O,\ D,\ M>$

(g) $< O, M, D, M >$ (h) $< M, M, D, M >$

Figure 3.2: Examples of navigation among moving obstacles

To allow the vehicle to safely pass through threatened areas, the prediction of moving obstacles should be incorporated into the route determination process. In certain circumstances (e.g., plumes), the affected roads could be temporally closed and be available again in the near future. Therefore, the waiting option can also be considered in the routing to minimize the total travel time, in the meantime avoiding the moving obstacles.

$< M, O, S, M >$ *Many moving objects have to be routed to One Static destination, avoiding many Moving obstacles*

A typical example of this situation is selecting $m$ fire trucks out of $n$ trucks ($m < n$) at different locations, and navigating them to one fire point, as illustrated in Figure 3.2b. Because the responders often work in groups, a reliable estimation of the time of their arrivals is very important in making emergency plans. The navigation system needs to take into account the information of vehicles (e.g., speed, departure time) and the road network (e.g., the affected time of roads) in order to provide a better estimation of arrival time. In some situations where there is only a limited number of roads to the destination, the system may need to be able to coordinate the paths of vehicles to avoid traffic congestion or possible collisions by considering other moving objects as obstacles.

$< O, M, S, M >$ *One moving object has to be routed to Many Static destinations, avoiding many Moving obstacles*

One classical example is navigating a rescue team to search several places within an area struck by disasters (see Figure 3.2c). The visiting order of these places are usually not pre-determined. So the navigation system must be able to plan a trip connecting these locations in an optimal or near-optimal way, taking into account the dynamic environment, which can be addressed as a dynamic version of TSP. Because the status of the road network affected by moving obstacles changes over time, the temporal aspects, such as the departure time, the operation time of emergency task performed at each location, etc., should also be incorporated into the routing process.

$< M, M, S, M >$ *Many moving objects have to be routed to Many Static destinations, avoiding many Moving obstacles*

This navigation problem can be considered as the Multiple Travel Salesmen Problem (MTSP), where more than one salesman is allowed to be used in the solution and many moving obstacles exist in the environment, as shown in Figure 3.2d. It has practical application in disaster management. When disasters (e.g., fires, plumes) occur, multiple rescue groups would be sent to a set of destinations to alert or rescue local citizens and each of them is responsible for a subset of the places. The problem is characterized by a dynamical environment and involvement of multiple destinations and multiple responders, which requires distribution of destinations among responders considering moving obstacles. Because situations may change unexpectedly or responders may fail to reach their destinations, the destinations need to be re-allocated to responders, which can be considered as a dynamic allocation problem.

$< O, O, D, M >$ *One moving object has to be navigated to One Dynamic destination, avoiding many Moving obstacles*

This is also a special case of pursuit-evasion problem. Figure 3.2e presents an example of this case that a police car has to pursue and stop a moving target (e.g., a thief, a suspect) in an area affected by moving obstacles (e.g., crowds, traffic jams). With existing monitoring technologies (e.g., camera, sensor), it is now possible to real-time track movements of the target and obstacles. More importantly, the models that can make reliable predictions of both the target and obstacles are also needed to estimate the best interception point.

*<M, O, D, M> Many moving objects have to be routed to One Dynamic destination, avoiding many Moving obstacles*

Figure 3.2f depicts such a situation that many police cars have to meet at a certain point or stop a moving suspect. The moving obstacles, which can either be hazards (e.g., floods, fires) or crowds, make parts of the road network temporarily unavailable, and the destination point changes with the dynamic environment. This situation can also be seen as variant of the PE problem, where multiple pursuers (responders) must coordinate their movements to jointly capture the evader, taking into account moving obstacles. Depending on real disaster situations, additional constrains can be applied to this problem. For example, all responders or a certain amount of them are required to arrival at the destination simultaneously. Similarly, in a multi-robot game, robots in a robotic team have to achieve a specific goal avoiding other opponent robots which can be viewed as obstacles. The pursuit strategies developed for robots could also be used by the navigation system to find the target point in the least possible time.

$< O, M, D, M >$ *One moving object has to be navigated to Many Dynamic destinations, avoiding many Moving obstacles*

This is similar to the aforementioned situation $< O, M, D, S >$, but the considered environment is more complex including multiple moving obstacles.

For example, in the case of tsunami, a response unit is sent to different rescue points in the field to guide people to the closest refuges. Because the sudden occurrence of disasters can cause panic among citizens and make people fleet in different directions, these rescue points varies with the changes of situations, as displayed in Figure 3.2g. The recent rapid advancement of sensor and communication technologies makes it possible to real-time monitor the situation in disaster events, which provides a promising solution to determine the rescue points for responders.

$< M, M, D, M >$ *Many moving objects have to be routed to Many Dynamic destinations, avoiding many Moving obstacles*

The case described in Figure 3.2h is the most complex one. Suppose that in an environment affected by floods, one ambulance has to meet other moving ambulances to transfer the wounded people to different hospitals. Not only the obstacles but also the states of vehicles change over time, which makes it quite difficult to have fixed meeting points. An efficient coordination between responders considering both the obstacles and vehicles is required to determine the optimal meeting points.

## 3.5 Investigation results

In this taxonomy, there are totally sixteen navigation cases. We investigated previous work on these navigation cases in the fields of emergency management and robotics. Table 3.3 summaries our investigation results. In the table, we list three aspects of interest, which are separately assigned to one column of the table. The column with the heading "Problem Type" gives the type of the navigation problem: shortest path problem (SPP), pursuit-evasion (PE) problem and traveling salesman problem (TSP). The next column with the heading "Environment Type" indicates whether any investigation has been conducted on the case corresponding to each listed environment type. The last column with the heading "Application Domain" tells if any relevant research on each navigation case have been studied in the concrete application domain. A tick $(\surd)$ denotes that it is under investigation, while a dash $(-)$ denotes no investigation is found and a dot $(\bullet)$ means uncertain. As indicated in Table 3.3, there are four navigation cases where typical shortest path routing is applicable. Totally eight navigation cases associated with dynamic one/more destinations can be seen as generalizations of the PE problem and have been intensively applied in robotics. The other four cases with multiple static destinations can be considered as the extended TSP problems. Although there are many methods developed for routing with obstacles, most of them are proposed in the mobile robot navigation domain and only a few approaches can be applied to obstacle-avoiding routing for emergency response. All in all, among the total sixteen navigation cases, to the best of our knowledge, only two of them, i.e., one moving object has to be navigated to one static destination, avoiding many static/moving obstacles ($< O, O, S, S >$, $< O, O, S, M >$), have been studied with consideration of constraints of road network by previous research. The remaining cases with obstacles should be further investigated, and many open issues relevant to navigation for first responders still need to be explored.

| | Navigation Case | Problem Type | Environment Type | | Application Domain | |
|---|---|---|---|---|---|---|
| | | | Road network | Free space | Emergency Response | Robotics |
| Navigation cases with static obstacles | $< O,O,S,S >$ | **SPP** | √ | √ | √ | √ |
| | $< M,O,S,S >$ | **SPP** | – | ● | – | ● |
| | $< O,M,S,S >$ | **TSP** | – | √ | – | ● |
| | $< M,M,S,S >$ | **TSP** | – | √ | √ | ● |
| | $< O,O,D,S >$ | **PE** | – | √ | – | √ |
| | $< M,O,D,S >$ | **PE** | – | √ | – | √ |
| | $< O,M,D,S >$ | **PE** | – | √ | – | √ |
| | $< M,M,D,S >$ | **PE** | – | √ | – | √ |
| Navigation cases with moving obstacles | $< O,O,S,M >$ | **SPP** | √ | √ | √ | √ |
| | $< M,O,S,M >$ | **SPP** | – | √ | – | √ |
| | $< O,M,S,M >$ | **TSP** | – | ● | – | ● |
| | $< M,M,S,M >$ | **TSP** | – | √ | – | √ |
| | $< O,O,D,M >$ | **PE** | – | √ | – | √ |
| | $< M,O,D,M >$ | **PE** | – | √ | – | √ |
| | $< O,M,D,M >$ | **PE** | – | √ | – | √ |
| | $< M,M,D,M >$ | **PE** | – | √ | – | √ |

Figure 3.3: Summary table of navigation taxonomy ($\surd$: it is under investigation; $-$: no investigation is found; $\bullet$: uncertain)

## 3.6 Proposed approach

To address the navigation problems involving the moving obstacles, in this research we propose our approach which combines the following technologies (See Figure 3.4):

- Hazard simulation models. We use hazard simulation models (e.g., fire model (Moreno et al., 2011), plume model (Zelle et al., 2013)) to provide

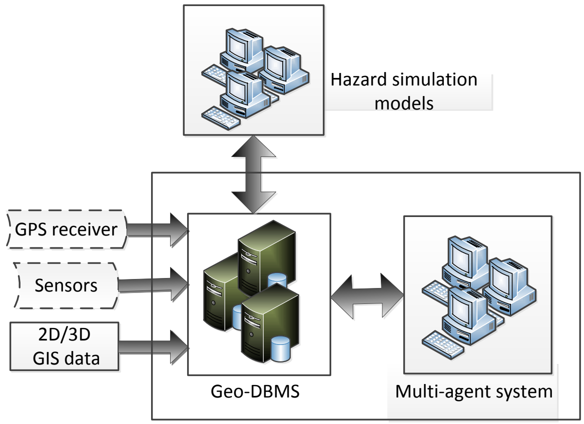



Figure 3.4: The overview of the generic system framework

predictions of obstacles caused by hazards. Using real-time sensor information (e.g., wind speed, air temperature, humidity, etc.) collected from the field, these hazard models can produce reliable predicted data about the movement of hazards, from which we derive the temporal information about the availabilities of road networks for routing.

- Geo-DBMS. In our research, a geo-Database management system (geo-DBMS) is selected and serves multiple purposes, including representation of hazard simulation results, management of spatio-temporal information of the road network, storage of the information regarding emergency tasks and relief vehicles (routes, source, destination, travel time, etc.), and supporting interoperability of the proposed system with other crisis management systems, etc.

- Multi-agent system. The agent technology is used to support the needed processing and analysis of spatial data. A set of software agents coupled with GIS functionalities is designed and developed to fetch forecast data of hazards from the database, transform the simulation results into the standard GIS format, and predict the state of roads in a certain area. Using predicted information about availabilities of roads, the path finding algorithms are employed by agents and applied to calculate the safe route avoiding moving obstacles.

The suggested routes, along with data of obstacles and vehicles, are presented to emergency managers in the control center and forwarded to mobile devices of responders on the field as well.

## 3.7 Concluding remarks

In this chapter, we have studied the following research question:

(1) What navigation cases need to be considered for assisting first responders?

From this formal description, we have introduced a taxonomy of navigation, which broadly classifies the navigation cases with obstacles, and have also briefly described possible approaches in addressing issues related to these cases. We have taken an overview of existing research on these navigation cases. According to our investigation, few studies have paid attention to navigation among obstacles in real road networks.

In this research, we aim at providing solutions for assisting first responders that are involved in the navigation cases with moving obstacles. As shown in the second half of Table 3.3, these are eight cases that involve moving obstacles. To start with, we focus on the cases with static destinations, since they occur more often and are simpler than the cases with dynamic destinations. Because navigation problems with both static obstacles and static destinations can be converted to well-known transportation problems and solved by traditional approaches, in this dissertation we limit our work to the cases that include only static destinations and moving obstacles. Specifically, we study the following four navigation cases:

(1) $< O, O, S, M >$ One moving object has to be routed to One Static destination, avoiding many Moving obstacles.

(2) $< M, O, S, M >$ Many moving objects have to be routed to One Static destination, avoiding many Moving obstacles.

(3) $< O, M, S, M >$ One moving object has to be routed to Many Static destinations, avoiding many Moving obstacles.

(4) $< M, M, S, M >$ Many moving objects have to be routed to Many Static destinations, avoiding many Moving obstacles.

# 4 DATA MODELS FOR PATH PLANNING AMONG MOVING OBSTACLES

In Chapter 3, we have given a taxonomy, which lists 16 navigation cases that first responders could confront during disasters. In this thesis, we aim at solving the following four navigation cases with moving obstacles and static destinations: 1) $< O, O, S, M >$; 2) $< M, O, S, M >$; 3) $< O, M, S, M >$; 4) $< M, M, S, M >$. To support path planning involved in these four cases, in this chapter we propose a series of data models to structure the data essential for routing. These data models capture both static information, such as the type of the response team, the topology of the road network, and dynamic information, such as sensor information, changing availabilities of roads during disasters, and the positions of the vehicles. In Section 4.1, we first give some discussions on different types of information needed for emergency navigation, followed by an introduction of geo-DMBSs to store this information in Section 4.2. Section 4.3 presents the general requirements that our data models should fulfill. Following these requirements, three versions of conceptual data models are built during the PhD study, and are described in Section 4.4, which gives a high level view of our data models. Section 4.5 describes the logical model corresponding to our selected model that contains all extensions, and gives its implementation details in the database management system. We draw some conclusions and end this chapter with some discussions in Section 4.6. This chapter is based on the following own publications: Wang et al. (2014), Wang and Zlatanova (2014), and Wang et al. (2015).

## 4.1 Information needed for emergency navigation

In this section, we present two types of information that are needed for emergency navigation: static and dynamic. Static information is relevant to topographic and territorial data (e.g., land use, road network, buildings, and locations of fire hydrants). Most of the static data can be obtained through municipality offices and the emergency response (ER) sectors, as well as public resources, such as the location of fire hydrants on www.openfiremap.org and general maps from OpenStreetMap (www.openstreetmap.org). Dynamic information is more related to the incident description and its impacts, damages, and sensor measurements, etc., and has a highly temporal aspect, i.e., it changes rapidly over time. This information consists of historic information, about what has happened since the disaster occurred, and predicted information, about what may happen. Examples of historical information are the type, scale, and affected area of an incident, the number of injured and missing people, etc. This information is needed to help emergency managers identify dangerous areas that should be avoided. Examples of predicted information are the likelihood of floods in a given 2.5-dimensional terrain, areas threatened by gas plumes, and the forecasted wildfire front, etc. Such information is also needed to assist planners in adjusting original route plans in advance of developing disasters.

## 4.2 Management of information in geo-DBMS

In this PhD study, a geo-Database Management System (geo-DBMS) is used to manage information related to emergency navigation. Geo-DBMSs have a couple of features that make it suitable for our work:

(1) Geo-DBMSs provide spatial data types (e.g., point, line, surface and volume) and functions that allow users to store, operate and manipulate geo-data, making the systems capable of handling different types of spatial data related to disasters;

(2) Special relationships and constrains can be defined and implemented in geo-DBMSs to check and to validate the spatial data, which ensure data consistency and data integrity. This is quite important for responders who need reliable data to perform their emergency tasks;

(3) Users can define their own datatypes and functionalities in geo-DBMSs, which allows them manage data produced in complex situations during disasters. For example, special datatypes with time component can be defined to represent the changing water depth in floods;

(4) Geo-DBMSs have mechanisms (spatial indexing and clustering) that enable fast update and access to geographic information. This would help facilitate quick response in large scale disastrous events, which often produce large spatial datasets;

(5) Geo-DBMSs have interfaces that permit applications outside the geo-DBMSs to access the data, supporting development of spatial function-

alities that are more advanced than the ones provided by geo-DBMSs. Furthermore, they allow multi-user access to the same database, which can contribute to generation of a common picture for responders;

(6) In carrying out emergency tasks, the responders may need not only spatial data, but also non-spatial data, such as the description of their tasks. Geo-DBMSs offer the capability of structuring and storing both spatial data and non-spatial data in a unified and consistent manner, making it suitable for management of the information required for most emergency tasks.

Generally, a data model is required to identify and organize data for a specific application of geo-DBMSs. In designing a data model, three steps can be distinguished: 1). identification of requirements, which is to specify the requirements the designed geo-DBMS should meet; 2). conceptual design, which is to create a conceptual data model that includes all essential entities as well as their attributes and relationships; 3). logical design, which is to translate the conceptual data model into the data model of a particular type of geo-DBMS (e.g., relational DBMS). In the following sections, we will describe how we develop the data model for emergency navigation during this PhD research.

## 4.3 Requirements for the data model

The development of data models for emergency navigation is supposed to satisfy a wide range of navigation cases presented in Chapter 3. Taking into account characteristics of the considered navigation cases, we derive a set of requirements that the data models should fulfill, and outline them as follows:

(1) Support representation of the environment, e.g., roads, junctions, and their relationship;

(2) Support dynamic simulation, such as the representations of disaster developments in time, changes in the availability of roads, and the movements of relief vehicles;

(3) Support dealing with uncertainties from hazard simulations;

(4) Support various spatial analysis, such as planning paths in the context of moving obstacles, coordinating routes for a group of responders, and identifying the areas that are most threatened, etc.;

(5) Support storage of the calculated results, e.g., the navigation route with one or multiple destinations, estimated traveling and arrival time;

(6) Should be compatible with the relevant data models for emergency response and existing standards defined by the Open Geospatial Consortium (OGC) or International Standard Organization (ISO), e.g., ISO 19107:2003 that provides a formal structure for representation of spatial objects.

## 4.4 Conceptual data model

According to the requirements listed above, we define three different versions (A, B and C) of spatial temporal data model to effectively organize all required information and knowledge in the geo-DBMS, using Unified Modeling Language (UML) class diagrams for database design. Basically the following 3 groups of data are considered: (1) data related to the road network; (2) data relevant to disasters; and (3) data on response units. The proposed models are designed adhering to the data model presented by Dilo and Zlatanova (2011) as much as possible, and have increasing capabilities. Using some classes from the model in Dilo and Zlatanova (2011), the first conceptual data model (CDM) is developed in Section 4.4.1 to capture the predicted information essential for navigation in the presence of moving obstacles. It is named CDM–A/Predictions. In Section 4.4.2, the second model is derived from the first one to handle the situations that involve multiple destinations. We call it CDM–B/Multi-destination. To enhance the routing capability of CDM–A and CDM–B, in section 4.4.3 we propose our third data model to support path planning among uncertain obstacles. We refer to the proposed data model as CDM–C/Uncertainty. We define the topology of the network by ourselves, and use the geometric data types specified by ISO 19107, e.g., GM_Point, GM_LineString, GM_Polygon, and GM_MultiSurface, to describe the spatial characteristics of geographic features. Because the data we are handling are constantly changing, new data types are created to capture this spatio-temporal nature.

### *4.4.1 CDM–A/Predictions*

Figure 4.1 is a UML class diagram presenting the conceptual data model CDM–A to structure data required for navigation among moving obstacles. The yellow classes are created for handling the data related to disasters. The light-green classes are used to support the representation of the road network. The classes in light-gray are defined for modeling the data of response units. New datatypes are colored in light-blue.

The core of model CDM–A is visualized in Figure 4.2. The class RoadNetwork is an extended graph, consisting of instances of RoadSegment that contain dynamic information produced by disaster events. To maintain the topology

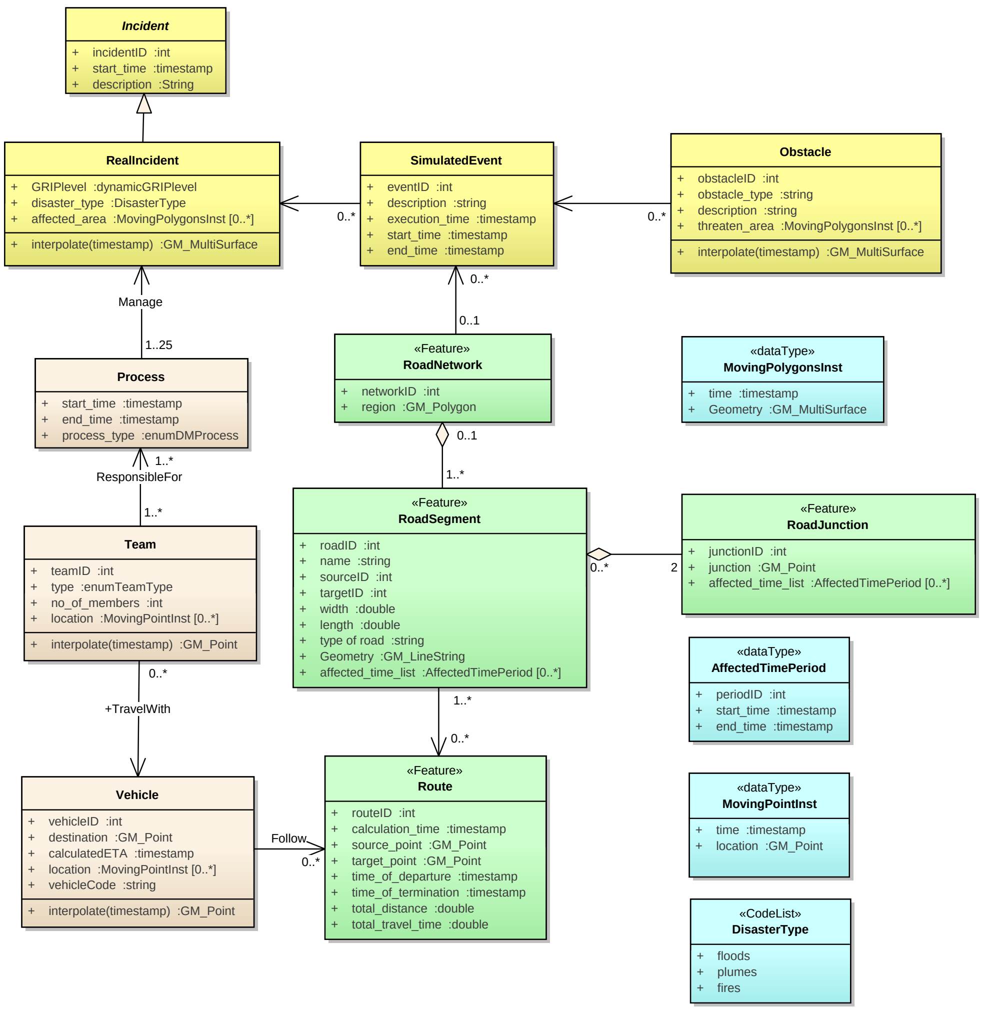



Figure 4.1: Conceptual data model CDM–A/Predictions, using ISO 19107 geometric data type (UML class diagram with classes related to disasters (in yellow), classes for representation of the road network (in light-green), classes relevant to response units (in light-gray), and newly defined datatypes (in light-blue))

of the road network, an aggregation association between RoadSegment and RoadJunction is established. Because each RoadSegment connects two RoadJunctions, the multiplicity at the side of RoadJunction is 2. Both RoadSegment and RoadJunction have an attribute affected_time_list used to store temporal information regarding the availabilities of the corresponding spatial objects. A new data type called AffectedTimePeriod is created for these two classes, containing the attribute of a dynamic nature.

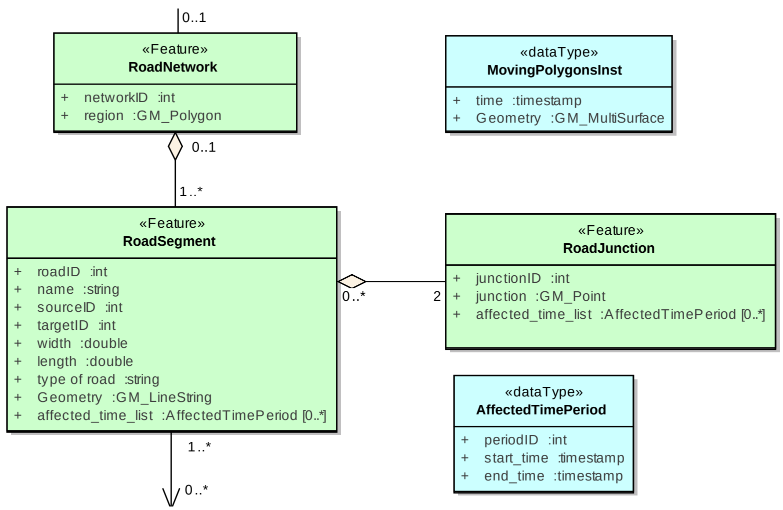



Figure 4.2: The core of model CDM–A/Predictions

A RealIncident is used to record the information of the disaster incident. It inherits all properties of the abstract class Incident which contains static information of the incident including incidentID identifying the incident, the location of the incident, the start time, and a text description of the incident. Some additional attributes are added to store the information generated during the incident, such as the disaster_type which can be floods, plumes, and fires, GRIPlevel describing the changing severity of the incident, and affected_area which stores the historic information of affected areas during the incident. The class SimulatedEvent is linked with RealIncident, and records the information related to hazard simulations. Following the ideas of bi-temporal model (containing both real world time and system time), we have two time dimensions maintained in class SimulatedEvent: 1). the time period for which the hazard simulations predict the effect of real incidents, which is represented by the attributes start_time and end_time; 2). the time when the hazard simulation is executed, which is stored in the attribute execution_time. In a SimulatedEvent, multiple obstacles can be produced. The class Obstacle stores the predicted formation of obstacles generated from the hazard simulations, and contains the following attributes: obstacleID, which is the identifier of the obstacle; obstacle_type, which is the type of hazards; description, which describes the characteristics of hazards, for example, water depth, the concentration of plumes; threaten_area, which records the geometric description of obstacles in the form of moving polygons.

To extract the temporal information of availabilities of the road network affected by moving obstacles, an intersection operation between the road network components (i.e., road segments and junctions) and the obstacle polygons with timestamp is performed. Here we make the following assumptions:

(i) If a road segment is temporarily not accessible, one or both road junctions connected to this road segment can still be accessible

(ii) If a road segment is partly not accessible, the whole road segment is considered inaccessible, as illustrated in Figure 4.3

(iii) If a road junction is not accessible, all road segments connected to this road junction are not accessible, as shown in Figure 4.4.

The obtained temporal information of the road network is stored in the attribute affected_time_list as mentioned earlier, and will be used by the algorithms presented in Section 6.2 and Section 6.3 to calculate the route avoiding the moving obstacles.

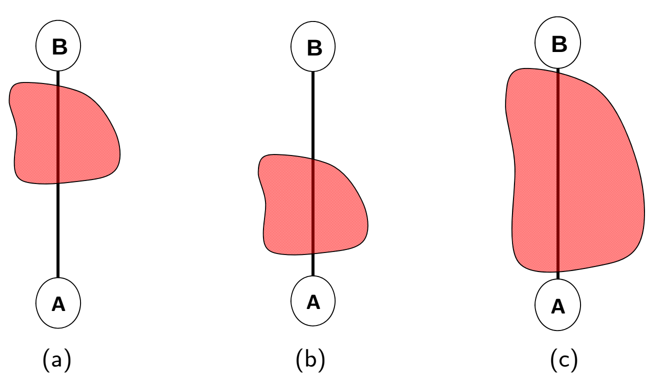



Figure 4.3: Intersection possibilities between a road segment and a obstacle (in polygons). Situations (a) and (b) are considered the same as situation (c).

After real incidents occur, different types of Processes, which are chosen from the 25 Processes defined within the Dutch law, are started to Manage the

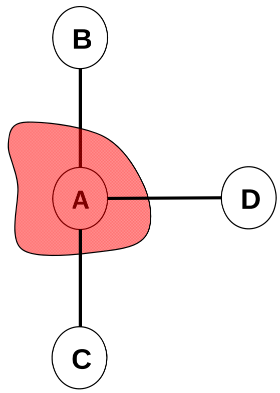



Figure 4.4: Intersection between a road junction and a obstacle (in polygons). The road segments $A\_B$, $A\_C$, and $A\_D$ are considered inaccessible.

incidents. Several Teams, who are responsible for handling these processes, are sent to response to the incident, using vehicles. The class Vehicle contains information related to vehicles. The association TravelWith captures the relationship between Team and Vehicle. The Routes, which are generated based on spatio-temporal information of the road network in the geo-DBMS, are proposed to the drivers. The association Follow is used to record the routes that drivers want to follow. The stored route information will also be used for monitoring movement of vehicles during disasters and analysed after disaster response.

### *4.4.2 CDM–B/Multi-destination*

As described in Chapter 3, in some navigation cases, responders need to go to multiple destinations to perform their tasks. With these considerations, we develop our data model CDM–B for storing the data about the relief vehicles and their routes. This data model is defined as an extension of the model CDM–A. The core of the CDM–B is shown in Figure 4.5.

Taken from the model in Zlatanova (2010), the class Task (highlighted in red in Figure 4.5) is used in our model to store the destinations of vehicles. It is linked with at most one Process that manages the RealIncident. Note the multiplicity at the side of Task is 1..*, which means that every process should be associated with at least one task. Each task consists of a set of operations that are described by the attribute description, and contains the information about the location where these operations should be performed. As mentioned in Diehl et al. (2006), time is critically important in the emergency response.

For some tasks, the responders have to arrive at the destination in a limited amount of time. For example, in the Netherlands, an ambulance must be in the incident place in 15 min (Wulterkens, 2007). To store this information, every task has an attribute required_arrival_time to indicate when the location associated with this task should be reached.

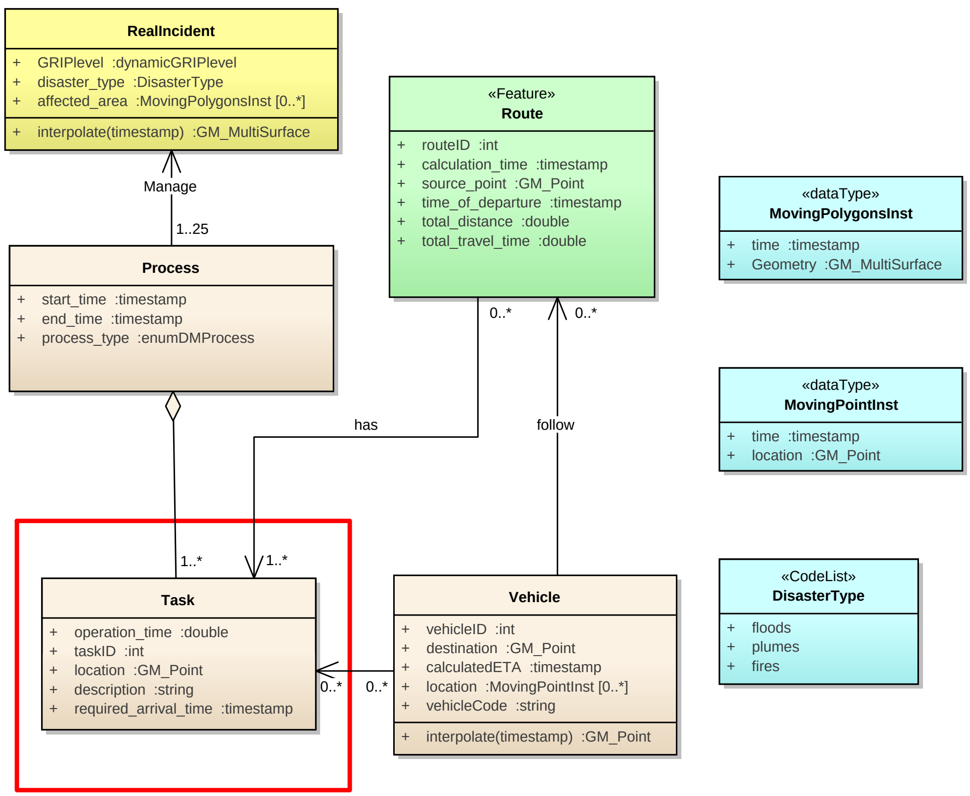



Figure 4.5: The core of data model CDM–B/Multi-destination

As mentioned earlier, in the case of big disasters responders need to go to a number of locations to perform tasks. Therefore a route of a rescue vehicle can be associated with a number of destinations. In model CDM–A, the positional information of destinations is explicitly stored in the Route. In CDM–B, this has been replaced by a relationship between Route and Task. It should be noted that 0..* is used at the side of Route, which means that Task can exist without linking with any Route. For example, for carrying out tasks, such as Report, RegisterInsident, no routes are required. On the other side, a Task can be associated with one or more routes for different vehicles, because in dealing with the tasks during big disasters, the cooperation and collaboration between responders are needed as well as different relief routes for them towards the same task. Besides, as responders differ on their capacities and responsibilities, they may need different amounts of time for performing the same tasks, which is represented by a many-to-many association between Vehicle and Task in the model.

### *4.4.3 CDM–C/Uncertainty*

In the early stage of the disaster response, the real-time information for the hazard models could be very limited, which results in uncertain results produced from the hazard simulations. With different settings of parameters, the hazard model could generate obstacles with different risks and shapes, which results in different impacts on the availability of roads. On the other hand, using protective equipment, the responders can still pass through some threaten roads if their risk of being affected by the obstacles is below a certain level. In this section, we provide a way to describe and quantify these uncertainties from the hazard simulation models, and design a data model to structure them to support routing.

Figure 4.6 shows the core of the mode CDM–C, which is defined for describing the uncertain moving obstacles and their influence on the road network. Covering the extension in CDM–B, the model CDM–C also supports routing for one or multiple responders towards one or multiple destinations. But it is enhanced with the capability of dealing with uncertainty of moving obstacles. In this model, we introduce the concept of levels to represent the state of the environment affected by the uncertain moving obstacles, instead of using binary values (i.e., closed and open).

In CDM–C, the class SimulatedEvent that is used in CDM–A and CDM–B is available again. However, in this model there is no direct relationship between SimulatedEvent and Obstacle, but only via a new class SimulationOuput (highlighted in red in Figure 4.6). Here the class SimulationOuput is introduced to differentiate and store multiple simulation results. It represents a specific execution of the simulation model, and links with a set of output data of Obstacles. A SimulationEvent can be associated with a number of SimulationOuput. Note that the input information for the simulation model is not included in the model, because the input variables vary in different simulation models and out of the scope of this research. But for a specific simulation model, this information can be easily organized in a separate class which can be linked to the SimulationOuput class.

The *risk level* is the main concept introduced in the model CDM–C to manage data about the uncertainty of components of the road network. In

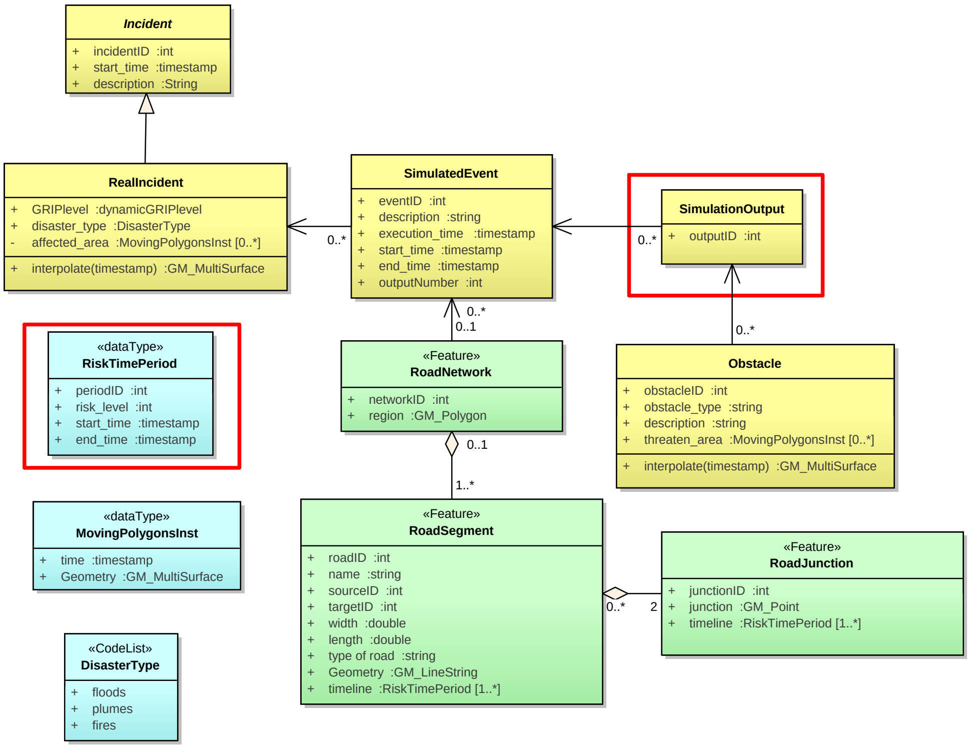



Figure 4.6: The core of model CDM–C/Uncertainty

CDM–C, the status of road segments and junctions affected by the uncertain moving obstacles is not represented by their availabilities (i.e., closed and open), but is described by a set of levels of risk. There are two situations that can be considered in describing the influence of the uncertain obstacles: 1). The obstacles have the same property (e.g., the same concentration of plumes, the same water depth) but uncertain geometries. Take Figure 4.7 as an example; 2). The obstacles have not only uncertain geometries but also different properties (e.g., different concentrations of plumes), as illustrated in Figure 4.8. To deal with the first situation, in this thesis we use the probability approach to quantify the uncertainty, and define the *risk level* according to the risk probability of intersecting with obstacle polygons. In our model, we do not explicitly store this risk probability, but discretize it by breaking the interval $[0, 1]$ into a finite number of smaller intervals. Each *risk level* corresponds to a specific risk probability interval. Take Figure 4.9 as an example, *risk level* $= L1$ corresponds to a range of risk probability $[0, 0.1]$, *risk level* $= L4$ to a range of risk probability $[0.4, 0.6]$, and so on. The larger the number in the *risk*

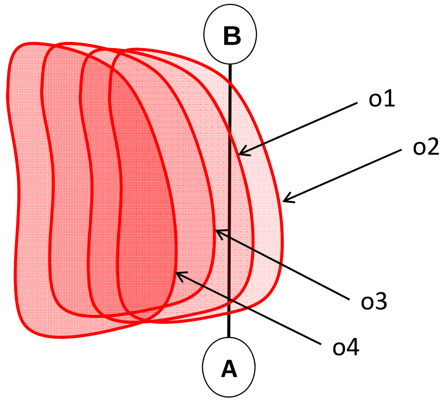



Figure 4.7: Different obstacles with the same concentration, caused by smoke plumes. The obstacles occur at the same simulation time, but have different locations and boundaries. Each obstacle corresponds to a specific execution of the plume model. The road segment $A\_B$ is affected by obstacles $o1$ and $o2$, and not affected by obstacles $o3$ and $o4$.

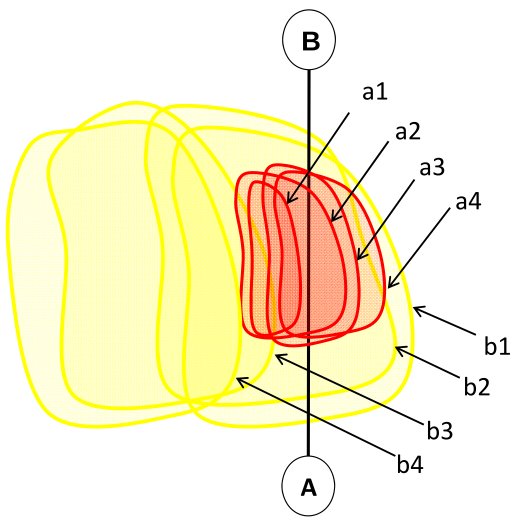



Figure 4.8: Different obstacles with different concentrations, caused by smoke plumes. The obstacles occur at the same simulation time, but have different locations and boundaries. The concentration of obstacles $a1$, $a2$, $a3$, $a4$ (in red) is 20 ppm (parts per million) or greater, and the concentration of obstacles $b1$, $b2$, $b3$, $b4$ (in yellow) is 0.5 ppm or greater. Obstacles $a1$ and $b1$ correspond to a specific execution of the plume model. The same holds for other obstacles. The road segment $A\_B$ is affected by obstacles $a2$, $a3$, $a4$, $b1$, and $b2$, and not affected by obstacles $a1$, $b3$, and $b4$.

*level* is, the more chances of exposure to the obstacles would be encountered. Regarding the second situation, the *risk level* is defined base on the combination of risk and risk probability interval. For instance, as shown in Figure 4.10, *risk level* $= L1$ corresponds to $\{20, [0, 0.1]\}$, where 20 is the concentration value of plumes, i.e., 20 milligrams per cubic meter $(mg/m^3)$, $[0, 0.1]$ is still the risk probability interval. Note that if a road segment or a road junction gets multiple *risk levels*, for example, $L4 = \{30, [0, 0.1]\}$ and $L7 = \{50, [0.3, 0.5]\}$, then the highest *risk level* will be assigned to represent its status. Using the concept *risk level*, the proposed model allows for a multi-state representation of the affected environment.

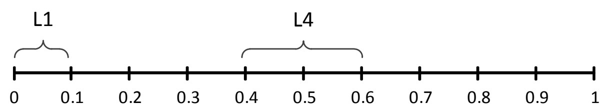



Figure 4.9: Defining of risk levels based on the risk probability

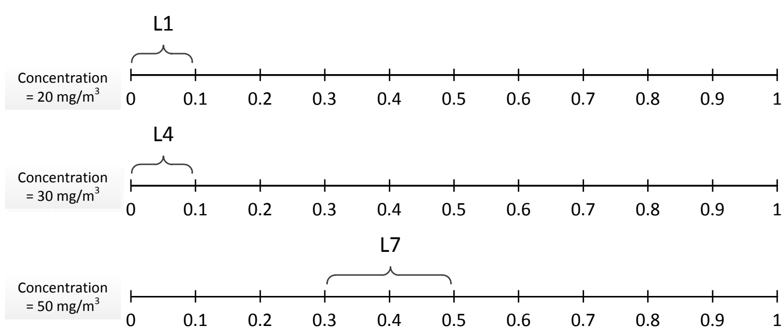



Figure 4.10: Defining of risk levels based on combination of the risk probability and the concentration

Because the uncertainties in hazard simulations are usually related to random variables or stochastic parameters, in this study we apply Monte Carlo (MC) method to obtain the risk probability of being affected by the uncertain obstacles. Given the uncertain variables of the simulation model as well as their probability, the process of the Monte Carlo approach for determining the uncertainty of being affected by the obstacles can be summarized as follows:

Step 1 Generate samples from $N_s$ realizations of the given random variables.

Step 2 Input these samples to run the simulation model, and generate $N_s$ simulation results.

Step 3 For each simulation result, determine whether the road segments and junctions are affected by checking if they intersect with the obstacle polygons.

Step 4 Count the number of times of being affected, i.e., $M_a$.

Step 5 Calculate the frequency of being affected by moving obstacles at a given time point, using the following formula:

$$P_b = M_a / N_s \tag{4.1}$$

This frequency $P_b$ is interpreted as an indicator of the likelihood of being affected by the obstacles. In our research, we use it as the risk probability to determine the *risk level* we introduced earlier.

To store the information of *risk level*, a new data type, RiskTimePeriod is created in CDM–C, as highlighted in red in Figure 4.6. Attributes of RiskTimePeriod are: periodID, which is the identifier of this time period; risk_level, which indicate the *risk level* during this time period; start_time, which is the starting time of the time period; end_time, which is the ending time of this time period. Using this data type, a new attribute timeline is added to replace the attribute affected_time_list in the classes RoadSegment and RoadJunction in models CDM–A and CDM–B, and to maintain information of the status of the road network at different time periods. In Section 6.4, we will illustrate how to use the *risk levels* to calculate the safe routes in the presence of uncertain moving obstacles.

### 4.5 Logical data model

The proposed conceptual data models have been translated into logical data models of a particular type of geo-DBMS. In our research, the object-relational database PostgreSQL with PostGIS extension (www.postgis.org) is used. PostGIS spatial data types and functions are compliant with OGC specifications and ISO 19107. Because all three conceptual data models are built with increasing capabilities and some classes are re-used in these models, it could cause confusions to present the logical data model for each version. As the model CDM–C can be seen as the successor to the CDM–A and CDM–B, in this section we elaborate on the logical data model fitting with model CDM–C.

Figure 4.11 shows the logical data model of CDM–C for PostGIS. Following classical approaches (Güting et al., 2000; Güting and Schneider, 2005), we create some new data types to store the spatio-temporal data, i.e., MovingPointInst to store dynamic positions of both vehicles and teams; MovingPolygonInst to

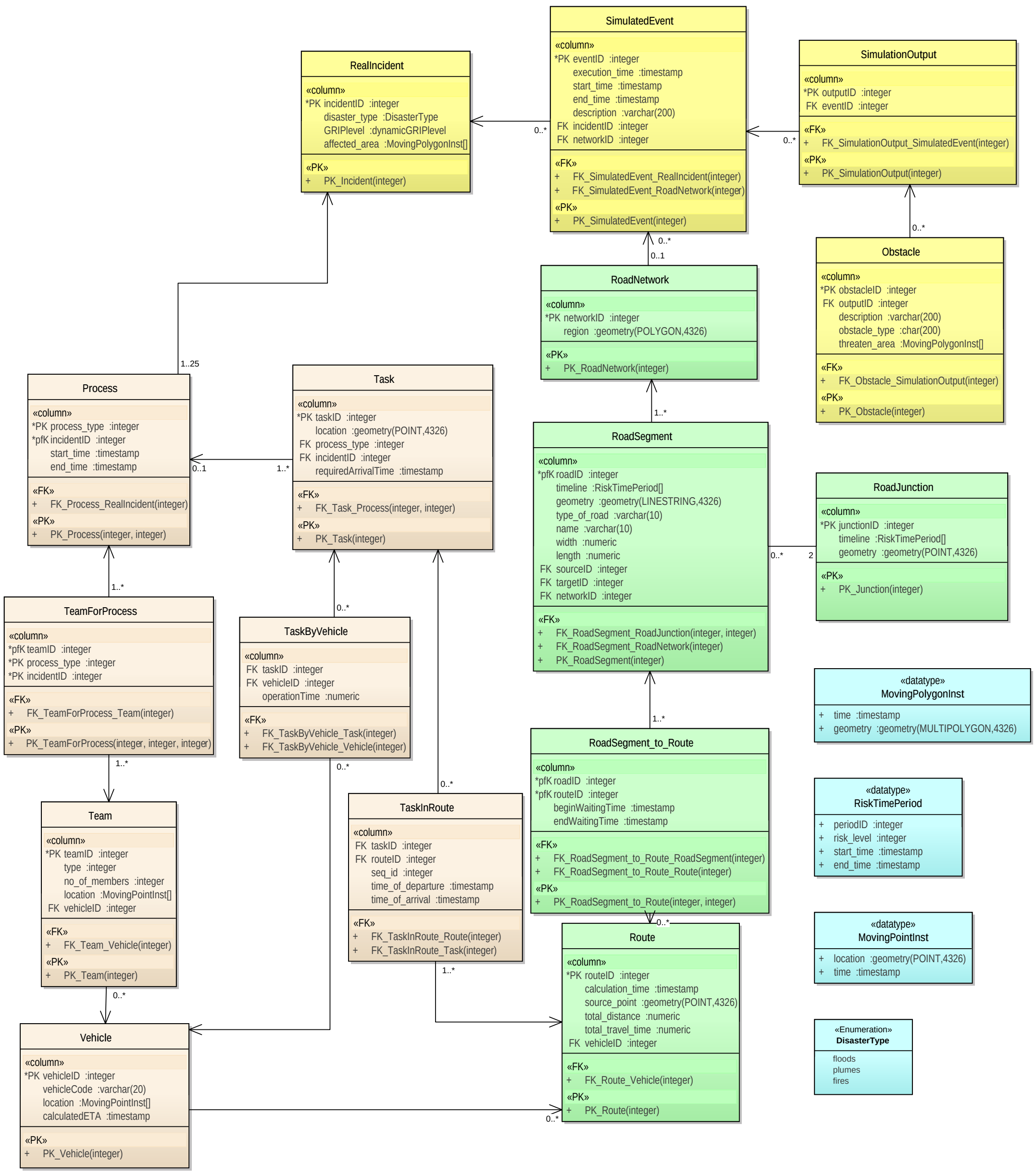



Figure 4.11: Logical data model fitting with model CDM–C (table diagram with PostGIS geometric data types, note that the ARRAY is used and indicated by square brackets [ ] after the datatype of the attribute)

record historic affected regions and identify dangerous areas in the near future. These data types are defined by adding timestamps as one of attributes to capture the temporal aspect. We use the ARRAY type, in which the new data types are used as a base type of the array elements, to record facts associated with time. For example, MovingPolygonInst[ ] is composed of a sequence of pairs of polygons and time instances. The temporal data stored in these arrays have different time resolutions according to needs of real applications. For example, the affected_area of a RealIncident is recorded with a time resolution of about 30 min; the movement data of the Obstacles has a time resolution of about 1 s for threaten_area. The interpolate methods are applied to these data if they do not have the required resolution.

To represent many-to-many associations, an intersection table is created. For instance, a table, RoadSegment_to_Route, is introduced to hold the many-to-many relationship between RoadSegment and Route, combining the primary keys from the original tables. TaskByVehicle is created by merging Task and Vehicle, and is added with an attribute operation_time to store the time required for the vehicle to carry out the tasks. The operation_time of tasks is updated by the responders in the field and will be used in the routing process. The classTaskInRoute is created to link Route and Task, and it has an attribute, called seq_id, to indicate the sequence number of the task to be visited by Vehicle.

```
CREATE TABLE Vehicle (
vehicle_id integer PRIMARY KEY,
vehicle_code varchar(20),
location MovingPointInst [],
calculatedETA timestamp);
```

(a) sql statement for creating table Vehicle

```
create Type MovingPointInst AS (
time timestamp,
location geometry(POINT,4326));
```

(b) sql statement for creating type MovingPointInst

```
CREATE TYPE DisasterType AS ENUM (
'floods',
'plumes',
'fires');
```

(c) sql statement for creating type DisasterType

Figure 4.12: The SQL statement generated from the UML model

The logical schema is automatically transformed by a modelling tool Enterprise Architect (www.sparxsystems.com) to a collection of Structured Query Language (SQL) scripts for creating and dropping tables. Figure 4.12a shows an example of SQL statements generated to create the table `Vehicle`. Because some tables use new data types, which can not be handled by Enterprise Architect in the automatic transformation, we manually implemented these special data types in PostgreSQL. For example, for the attribute `location` in table `Vehicle`, we created the data type `MovingPointInst`, as shown in Figure 4.12b. Other data types are also created and added into the SQL scripts, such as the enumerated type `DisasterType` (see Figure 4.12c).

These created tables are populated with spatial and spatio-temporal data that are used for analysis and visualization by our navigation system as well as traditional GIS tools.

## 4.6 Concluding remarks

In this chapter, efforts have been directed towards addressing the following research question:

(2) What data models should be developed to support path planning among moving obstacles?

As hazards with some physical phenomena can have impacts on the availabilities of the road network, this information is important for disaster response and need to be structured in geo-DBMS to support path planning among moving obstacles. In our geo-DBMS, a set of spatio-temporal data models for the management of both static and dynamic disaster-related information, considering the type of disasters, the number of responders, the uncertainties of hazard simulation results, and so on. On the basis of our data models, the geo-DBMS, which is updated constantly, can provide latest and most consistent data required for navigation during specific disasters.

It should be noted that, although the hazards we consider in this chapter are forest fires and toxic plumes, the developed approach is not limited to theses hazardous events. Our central goal in this PhD study is to provide safe paths among obstacles caused by various disasters, e.g, floods, landslides, explosions. The approach introduced here can be tailored for other types of disasters. For example, in the designed data model, obstacles caused by floods can also be represented as moving polygons.

# 5 A MULTI-AGENT BASED NAVIGATION SYSTEM ASSISTING FIRST RESPONDERS AMONG MOVING OBSTACLES

The previous chapter (Chapter 4) focuses on the structuring and storage of data essential for emergency navigation, which enable our geo-DBMS to provide the needed and consistent data for navigation systems. In connection with the geo-DBMS, a multi-agent based navigation system, which supports the processing and analysis of the provided data, is presented in this chapter. In Section 5.1, we start with analyzing the general requirements for building an emergency navigation system. Section 5.2 discusses the motivation of use of agent technology in support of navigation for first responders. In section 5.3, we illustrate the architecture of the proposed multi-agent based navigation system, which contains different types of software agents with specific functions. The chapter ends with some conclusions in section 5.4. This chapter is based on the following publications: Wang and Zlatanova (2013b) and Wang and Zlatanova (2014).

## 5.1 General requirements of system design

In this research, we aim to design and develop a navigation system, which is capable of assisting emergency managers in the command and control center in planning routes for responders among moving obstacles. The building of such a navigation system involves the necessity of reflecting on functions that the system should provide. Taking into account characteristics of the considered navigation cases, in this section we derive a set of functional requirements for the proposed system and outline them as follows:

(1) Should support data fusion and transformation. The system should be capable of integrating multiple spatial datasets in different format from different data sources, and transforming them into a uniform format;

(2) Should support data processing. The system needs to process the data about hazards, and to extract useful information and knowledge essential for emergency navigation;

(3) Should support route planning. It should not only assist routing for individual responders as well as for multiple responders, but also be able to autonomously recalculate routes in response to environment changes;

(4) Should support storage and representation of spatio-temporal information. Not only the static information (e.g., the topology of road network) but also the dynamic information (e.g., the shape of obstacles, the position of vehicles) should be represented and stored in the database for navigation purpose;

(5) Should support visualization of the movement of vehicles and hazards. This is useful for emergency managers to enhance situational awareness as well as to evaluate the route planning;

(6) Should be compatible with the emergency response procedures. The system should be able to assist emergency actors in performing tasks to facilitate the emergency response.

## 5.2 Using the agent technology

In this research, the agent technology is applied to build our navigation system to support path planning among moving obstacles, assisting emergency actors in performing their tasks. There are three main reasons that make us believe that the agent technology is a proper tool for supporting navigation among moving obstacles:

*1. Supporting distributed control*

The path planning among moving obstacles is a complex process, consisting of a number of sub-tasks (e.g. data collection, data processing, and route generation). By decomposing a large system into smaller components, the agent technology provides a distributed architecture, in which these tasks can be distributed to a group of agents and each agent coupled with specific skills is responsible for handling and controlling a set of specialized tasks. This decentralization reduces the complexity of developments of the system, and allows distribution of the computing resources for fulfilling the tasks. What's more, the agent technology provides the communication facilities that enable agents to communicate with each other. This allows the tasks of agents to be coordinated to achieve the global goal, i.e., providing safe routes for responders to their destinations. Especially when the datasets for emergency navigation are stored and located in different machines and in different places, using the predefined communication protocol, agents can be deployed across different platforms, monitoring the dynamic environments and sending messages to trigger the tasks (e.g., planning or re-planning routes) of other agents.

*2. Supporting route calculation*

The navigation problems we are dealing with involve multiple responders and multiple destinations, which can be considered as the NP-hard problem. The complexity of these problems is increased when the moving obstacles have to be considered in the routing. The multi-agent systems provide a promising approach to these problems, by distributing the route computation to multiple individual agents. Within the multi-agent system, each individual agent focuses on its own calculation and plans a route to a subset of destinations, and all these partial route plans are evaluated, selected, and combined to generate an overall route plan assigning all destinations to multiple responders. Furthermore, responders have different profiles (e.g., the speed, protection equipment, maximal number of delays), which requires different constrains that should be imposed to route generation for the responders. By representing each emergency actor in real disasters as an agent, the profile of responders can be easily converted into the agent model, configuring the parameters and preferences of agents, which supports the generation of routes customized for the responders. More importantly, the agent system has autonomous capabilities that can support quick and automatic spatial data processing and analysis. This is quite important for the disaster response because disasters often change fast over time. Using autonomous behaviors, the agents can constantly respond to situation changes without users' participation, generating alternative routes automatically according to new situations.

*3. Allowing for flexibility and adaptability*

Because the environment could be affected by disasters that are not foreseen during the development of the system, under the same framework, agents with specific knowledge of these hazards can be developed, and be easily added into the system, transforming the output data from these hazard simulations into a desired and uniform format. The agent technology also supports integration with other crisis management systems. The navigation system could have to be connected with other crisis management systems that are unknown during the design phase. These crisis management systems may have different data formats, different interfaces, different communication protocols, and different degrees of accessibility to internal functionalities. Confirming to the same software agent structure, new types of agents can be implemented and embedded into the system, without requiring the re-design and re-development of the whole software architecture.

## 5.3 The architecture of the multi-agent based navigation system

Taking into account the requirements presented in Section 5.1, we design and develop a multi-agent based navigation system for first responders among moving obstacles. We mainly focus on the design and development of agents (definition of roles, functionalities, interactions etc.). Aspects regarding hazard simulation models are out scope of this dissertation.

The architecture of the multi-agent based navigation system consists of four modules (see Figure 5.1):

(1) Prediction module, which contains Hazard Agents that are associated with different hazard models producing the predicted data about moving obstacles;

(2) Monitoring module, which consists of Task Monitoring Agent, Network Monitoring Agent, and Task Monitoring Agent, supporting the monitoring of the road network, vehicles, and tasks;

(3) Path planning module, which is composed of two types of agent: Task Allocation Agent and Vehicle Agent, and is responsible for generation and coordination of routes;

(4) Visualization module, which provides 2D and 3D visualization of routes as well as the environment.

In the prediction module, monitoring module, and path planning module, each type of agents has its specialized skills and performs the operations involved in emergency navigation, assisting responders in carrying out emergency tasks. The visualization module, which contains applications that display maps using 2D and 3D geo-information, is a supporting component for the agent system. This multi-agent architecture allows the system to be distributed at different machines, each of whom contains one or more types of agents running concurrently and communicating via the pre-defined protocol. It also permits the system to scale easily and brings the flexibility in future implementation of more functionalities. For example, a user interface agent can be added into the system to display the map according the profile of users.

In the following sections, all modules and related agents will be described, with the focus on the monitoring module and the path planning module.

### *5.3.1 Prediction module*

The prediction module is used to integrate data from mathematical models of different types of hazard and to provide the predicted information of moving obstacles for the route planning. In this module, different types of Hazard Agents are extended from the generic Hazard Agent and developed for handling and operating data from different hazard simulators. As mentioned earlier, because emergency managers are usually not either hazard researchers or GIS experts, this module can be used to help emergency managers understand the future situations, and supports them in decision-making process for managing different disasters, analyzing the hazard simulation results.

*Hazard Agent*

The Hazard Agent is an abstract agent from which specialized agents for specific types of hazards can be derived. It enables integration of heterogeneous data from various hazard simulators. Specifically, the Hazard Agents are customized with specific knowledge of hazard simulations, and convert the hazard simulation results to the polygons with different timestamps to represent obstacles that block parts of the road network (see Figure 5.2). For example, in the case of forest fires, a fire agent can be created to collect data on fire-affected units from the fire simulation and to produce the polygons that represent the fire spread. Because in real disasters, multiple types of hazards may occur and affect the road network simultaneously, different types of hazard agents would be needed, and integrated into the module to work together, processing the data from different hazard simulation models. Besides, it also informs the other modules about new predictions generated by hazard simulations, which allows re-planning of routes if the current plan cannot be carried out due to situation changes.

### *5.3.2 Monitoring module*

This monitoring module consists of three types of agents: 1). The Task Monitoring Agent, which helps the command and control center to collect and check emergency tasks; 2). The Network Monitoring Agent, which handles the spatio-temporal information of the road network; 3). The Vehicle Monitoring Agent, which performs real-time tracking of the moving vehicles This module makes a direct connection to the geo-DBMS that is updated by the hazard simulations and by the responders on the field, notifies other agents of

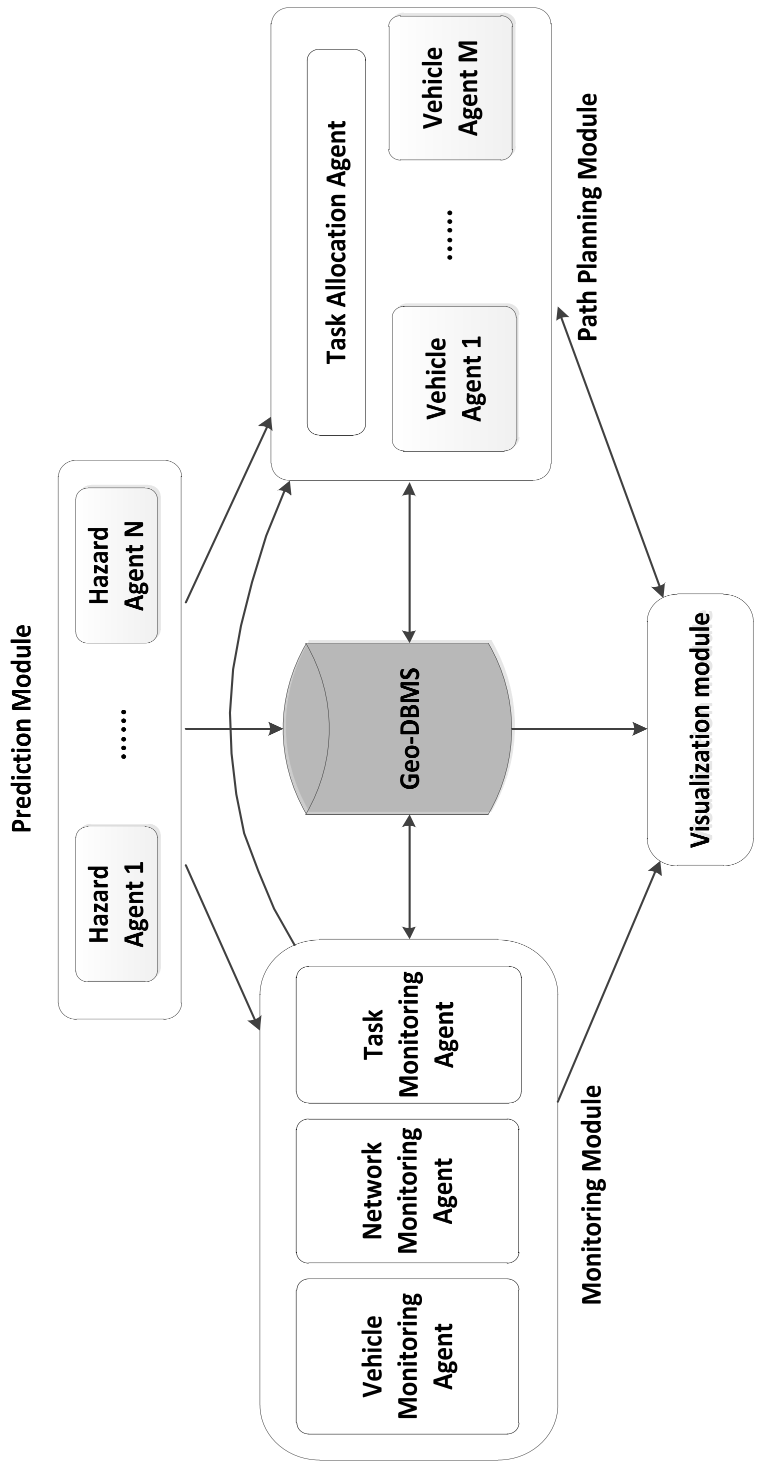
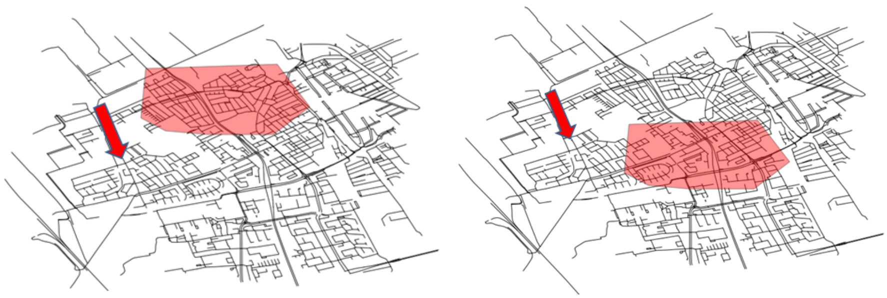



Figure 5.1: The architecture of the proposed multi-agent based navigation system



(a) T=$t_1$ (b) T=$t_2$

Figure 5.2: Snapshot of a moving obstacle at different time instances ($t_1 < t_2$)

environment changes in affected areas, and provides situational information that is further analysed by other modules.

*Task Monitoring Agent*

The Task Monitoring Agent constantly checks the state of all tasks stored in the database. After occurrence of disasters, the command and control center collects both dynamic and static data about the incident, such as the location of incidents, how many people are injured, how many responders are needed, etc., and generates tasks for the emergency response units to carry out on the field. All these data are structured and stored in the database to support routing, according to our defined data model (see Chapter 4). The Task Monitoring Agent assists the command and control center in checking the stored tasks to see which tasks are fulfilled or not. Moreover, it will also send the uncompleted tasks to path planning module (see Section 5.3.3) for planning or re-planning of routes, if a route calculation request is received.

*Network Monitoring Agent*

The Network Monitoring Agent collects spatio-temporal information of the road network, using the predicted data of hazards. When new predictions of hazards come, it first retrieves the data of obstacles and the road network from the database. Then it performs an intersection operation between the obstacle polygons and the roads, determining all the affected roads and the time intervals when they are closed or open. It is also responsible for updating the collected information into the database. After the update is finished, it immediately sends a request of re-planning routes to the agents in the path planning module, using the communication facilities.

*Vehicle Monitoring Agent*

The Vehicle Monitoring Agent constantly keeps track of the relief vehicles in real time. It uses GPS data to analyse the state of moving vehicles. If the vehicle is found to be trapped by traffic jams, or not be able to fulfill its task, the Vehicle Monitoring Agent will send the list of all available vehicles to the path planning module, and suggest a re-planning of routes for the other vehicles to fulfill the tasks, using the pre-defined communication protocol. It also provides the real-time information of vehicles (such as the current position, the status, the routes, and so on) for visualization purposes.

### 5.3.3 Path planning module

This path planning module makes a central planning of routes and broadcasts them to the vehicles. The routes are generated by means of distributed mechanism implemented by MAS that consists of two types of agent: Task Allocation Agent and Vehicle Agent. The agents work in an integrated way to achieve their common goal, e.g., minimizing the total traveling time over all vehicles, while focusing their individual interests, e.g., optimizing the route carrying out the allocated tasks.

*Task Allocation Agent*

The task allocation agent is mainly responsible for the coordination of all rescue vehicles and tasks involved in the disaster response. It allocates tasks to the Vehicle Agents, using a dynamic allocation strategy. When a route calculation is requested by the user or other modules of the system, the task allocation agent first communicates with the Task Monitoring Agent for tasks that have not been fulfilled in current situations, and obtains the information about the available vehicles from the Vehicle Monitoring Agent. Then the task allocation agent uses the auction algorithm (see Section 6.6 for more details) to decide which vehicles to go to which destinations, based on the bids with travel cost estimated by the Vehicle Agents (as seen in Figure 5.3). After the calculation is completed, it also forwards calculated results to the mobile devices of real rescue units.

*Vehicle Agent*

The Vehicle Agent is associated with a corresponding rescue unit (e.g., a fire truck, a police car, and an ambulance), and is characterized by a set of properties (e.g., position, current speed, maximal speed, and type of vehicles). During task allocation, it receives the tasks sent by the Task Allocation Agent, fetches

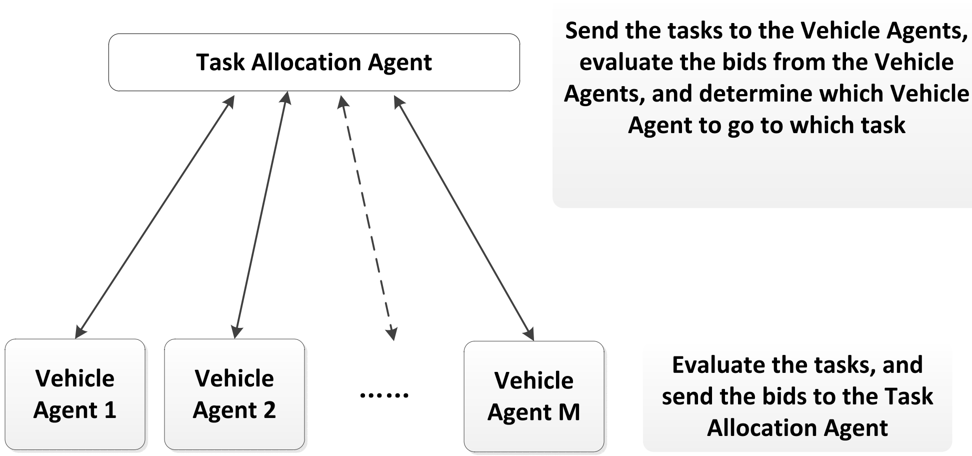



Figure 5.3: Interaction between Task Allocation Agent and Vehicle Agents

the real-time positions of the vehicles, computes the cost of carrying out the new tasks, and bids on the new tasks. The bid, together with the calculated results of routes (e.g., travel time, completion time of task, travel distance, etc.), is forwarded to the Task Allocation Agent for evaluation, as shown in Figure 5.3. In our system, the Vehicle Agent estimates completion time of the new tasks, considering the departure time, the vehicle speed, and the predicted information of the road network provided by prediction module. It generates routes using the routing algorithms presented in the following Chapter 6.

### *5.3.4 Visualization module*

The visualization module is developed to help emergency managers to better understand the current and future situations. This module can run on a local desktop computer and allow users to check the predicted information of moving obstacles, select destinations for vehicles, and evaluate route plans. We have developed a 2D viewer in this module, using OpenStreetMap as a base map. This 2D viewer visualizes the regions where first responders work and provides an overview of planned paths. More importantly, it also supports animation of predicted movement of hazards and vehicles. The viewer can display the movement of obstacles that cross the road network, and activities of the agents that represent the responders who move and carry out their tasks, from which the situational awareness of emergency managers can be enhanced.

## 5.4 Concluding remarks

In this chapter, we have presented a multi-agent based navigation system to provide solutions to the different types of navigation problems, addressing the following research question:

(3) What types of agents are needed to assist path planning among moving obstacles?

Six types of agents are designed and developed in the proposed system assists emergency actors in data collection, data processing, route generation, and task allocation. Such a multi-agent based system supports connection of other software and applications, for example, hazard simulation models which provide predicted information of obstacles, network analysis applications that contain path planning algorithms for calculation of obstacle avoiding routes, etc. Furthermore, using the agent-based simulations, the system provides a visualization tool that can display "future" situations with the current and predicted information. This makes it possible for responders to evaluate the route plans for disaster response, and to modify them if necessary, which contributes to improvement of situational awareness and decision making.

# 6 ALGORITHMS FOR PATH PLANNING

In Chapter 5, the design of a multi-agent based navigation system for first responders has been presented. In this chapter, particular attention is paid to the core of the navigation system, i.e., path planning algorithms that use the provided data to calculate routes for responders among moving obstacles. Basically, three types of path panning problems are considered: one-to-one, one-to-many, and many-to-many. Because traditional shortest path algorithms, which are mainly developed for routing in static environments, can not be directly applied to the road network affected by moving obstacles, special algorithms are needed to deal with dynamic information of roads. In Section 6.1, we first provide an overview of the basic principle of the A* algorithm. Based on the A* algorithm, three versions (I, II and III) of one-to-one path planning algorithms are developed for dealing with different situations. In Section 6.2, we focus on the route planning in the presence of obstacles like forest fires, and present a Moving Obstacle Avoiding A* (MOAAstar) algorithm, which does not consider the waiting option and uses the predicted availabilities of roads to generate routes avoiding the obstacles. It is called MOAAstar–I/Non-waiting. Section 6.3 describes another A*–based algorithm, which can incorporate waiting options to avoid moving obstacles. We name it MOAAstar–II/Waiting. For the situations that involve uncertain obstacles, in Section 6.4 we provide our third MOAAstar algorithm, considering both the uncertainty of moving obstacles and the profile of responders. We refer to the proposed algorithm as MOAAstar–III/Uncertainty. Section 6.5 outlines a one-to-many path planning algorithm that takes into account multiple parameters (e.g., the time required to carry out the task, the speed of vehicles, and departure time). In Section 6.6, we provide an auction-based approach which allows the system to allocate different target locations to the responders. In Section 6.7, we conclude this chapter with some discussions on the developed algorithms. This chapter is based on following three publications: Wang and Zlatanova (2013a), Wang et al. (2014), and Wang and Zlatanova (2014).

## 6.1 A* algorithm for one-to-one path planning

In this section, we give an brief introduction of the A* algorithm, illustrating the basic principle of the A* algorithm.

A* is a well-known algorithm developed to solve the one-to-one shortest path problem (Hart et al., 1968). The A* algorithm uses a heuristic function to estimate cost from each node to the destination to guide path search. The cost associated with a node $x$ is $f(x) = g(x) + h(x)$, where $g(x)$ is the actual cost of the path from the start to node $x$, and $h(x)$ is an estimated cost from node $x$ to the destination, e.g., the Euclidean distance between $x$ and the destination. Figure 6.1 shows the structure of the standard A* search algorithm. As shown in the figure, the algorithm maintains two important sets: *openSet* that stores nodes who are not expanded, and *closedSet* that stores nodes who have been expanded. At each iteration, the algorithm selects node $x$ with the minimal cost from the *openSet* for expansion. All successors of node $x$ that are unexplored will be put in the *openSet* for further expansion.

**A* algorithm**

```
Initialize source S, destination D,
openSet = ∅, closedSet = ∅
insert S in openSet
while openSet is not empty do
    x = the node in openSet having the lowest f value
    if x = D then
        return the path from S to D
    end if
    remove x from openSet
    insert x to closedSet
    for each neighbor y of x that is not in closedSet do
        if y is not in openSet then
            insert y into openSet
        else if g(x) + l_xy < g(y) then
            update the backpointer of y to point to x
        end if
    end for
end while
return no-path
```

Figure 6.1: A* search algorithm

## 6.2 Algorithm MOAAstar–I/Non-waiting for one-to-one path planning

In this section, we focus on the routing process in the case of forest fires, and present our first extended A* algorithm, Moving Obstacle Avoiding A*

(MOAAstar) algorithm, which does not use the waiting option and supports finding paths avoiding moving fires. It is called MOAAstar–I/Non-waiting. The modified algorithm uses the data from the geo-DBMS built on the data model CDM–A/Predictions, and calculates obstacle avoiding route from a given starting time, aiming to minimize the total travel time.

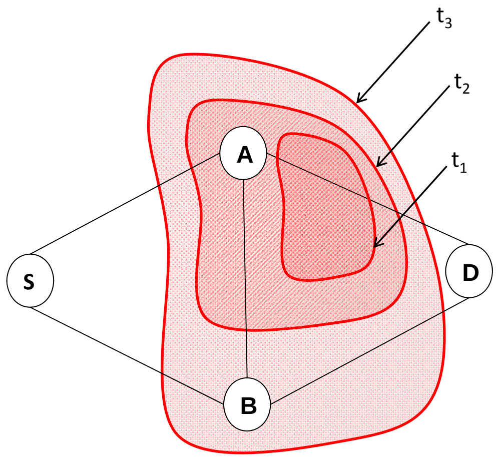



Figure 6.2: A simple road network affected by fires (in red polygons). At time $t_1$, road segment $A\_D$ is not available. At time $t_2$, road segments $A\_B$, $A\_D$ and $A\_S$, and road junction $A$ are not available. At time $t_3$, road segments $A\_B$, $A\_D$, $A\_S$, $B\_S$ and $B\_D$, and road junctions $A$ and $B$ are not available ($t_1 < t_2 < t_3$)

Let $G = (N, E)$ be a directed graph consisting of a finite set of nodes $N$ and edges $E$. Each node $x \in N$ corresponds to an object of class RoadJunction, and each edge $xy \in E$, connecting from node $x$ to node $y$, corresponds to an object of class RoadSegment. We use $l_{xy}$ to represent the length of edge $xy$. Here we assume that once the nodes and edges are affected by the fires, they will not be available anymore (as illustrated in Figure 6.2). Following this assumption, every affected edge has only one AffectedTimePeriod stored in the affected_time_list. The start_time of the AffectedTimePeriod is the time when the edge is affected by the fires, and we use $t_c^{xy}$ to represent it for edge $xy$. The end_time of the AffectedTimePeriod is set to $inf$ by default.

In MOAAstar–I, we take into account the affected time of road segments and do not consider the information on the state of road junctions. This is because that waiting would not be safe during fires and the vehicles need to move as fast as possible, according to the assumption mentioned above. In the algorithm MOAAstar–I, we introduce an additional parameter for the algorithm, the speed of vehicles $speed$, to select nodes for expansion. The value of $speed$ can be obtained in two ways: (1) user configuration; (2) real-time calculation

**MOAAstar–I/Non-waiting**

```
Initialize source S, destination D, speed, departureTime
Initialize openSet, closedSet
g(S) := departureTime
Insert S in openSet
while openSet is not empty do
  x := the node in openSet having the lowest f value
  if x = D then
    return the path from S to D
  end if
  Remove x from openSet
  Insert x to closedSet
  for each neighbor y of x do
    if y in closedSet then
      continue
    end if
    tentative_cost := g(x) + l_xy/speed
    flag := false
    if y not in openSet then
      if tentative_cost < t_c^xy then
        Insert y to openSet
        flag := true
      end if
    else if (tentative_cost < g(y)) and (tentative_cost < t_c^xy) then
      flag := true
    else
      flag := false
    end if
    if flag = true then
      the backpointer of y := x
      g(y) := tentative_cost   /* the actual path cost from S
to node y */
      h(y) := Euclidean_distance(y, D)/speed
      f(y) := g(y) + h(y)
    end if
  end for
end while
return no-path
```

Figure 6.3: The main structure of MOAAstar–I/Non-waiting

based on the location of vehicles recorded in the database. A new parameter *departureTime* is added to help estimation of arrival time of each node. Figure 6.3 shows the main structure of the algorithm MOAAstar–I. When a node $x$ is expanded, we compute the estimated arrival time considering the cost of the edge $l_{xy}$ and the given *speed* (see line 16). At line 19, we use a condition to decide if the successor $y$ of $x$ should be added to the *openSet*. If the object can safely pass through the edge between the expanded node $n$ and the successor $y$, i.e., the estimated arrival time is earlier than the closed time of the edge $t_c^{xy}$, the successor $y$ will be added into the *openSet* for further expansions. If not, it remains un-explored. The same condition is also applied on line 23, which guarantees that the evaluated node $y$ should be updated not only with the faster arrival time but also with the safety of passing through the edge $xy$.

*Theoretical analysis*

Here we sketch the proof of the optimality of the path calculated by our algorithm.

**Theorem 1.** *When the modified A* selects the goal for expansion, it has found a time-minimal and safe path to the destination node D.*

*Proof.* Were this not the case, the optimal path, $P$, must have a node $x$ that is not yet expanded (If the optimal path has been completely expanded, the goal would have been reached along the optimal path.). There are then the following two possibilities resulting in the fact that $x$ is not expanded to generate successors: (1) $f(x) > f(D)$; (2) all successors of $x$ cannot be safely reached, i.e. the estimated arrival time is after the closing time of the edge between $n'$ and its successor. Because $f$ is non-decreasing along any path, $x$ would have a lower $f$-cost than $D$ and would have been selected first for expansion before the goal node, which contradicts the first possibility. We assume $y$ is the successor of $n$ along the optimal path, implying that $g(x) + l_{xy}/speed < t_c^{xy}$, which eliminates the second possibility. In the algorithm, the cost on an edge is equal to the time it takes to execute that edge, and whenever a $g$-value is updated (a shorter path is found), the time value is also updated to the earlier time. Therefore, when the node $D$ is expanded, it is the earliest time we can arrive at the goal node. This is optimal in terms of time cost. We also know that all explored nodes are safely reached, which makes the entire path safe, from the source node to the destination node.

## 6.3 Algorithm MOAAstar–II/Waiting for one-to-one path planning

As described in chapter 1, in some circumstances the moving obstacles (e.g. plumes) can make roads unavailable for some specific periods of time. In this case, waiting at some points strategically might be the fastest and safest option. For example, in a road network as shown in Figure 6.4, a fire truck has to go from point S to point D. A toxic plume moves across a road segment and blocks the vehicle's way when it arrives at grey point. If the moving obstacle can move away from the determined route soon, the fire truck can wait for only a short period of time, and continues its way to reach the destination point, saving more time than it would take to follow other alternative routes.

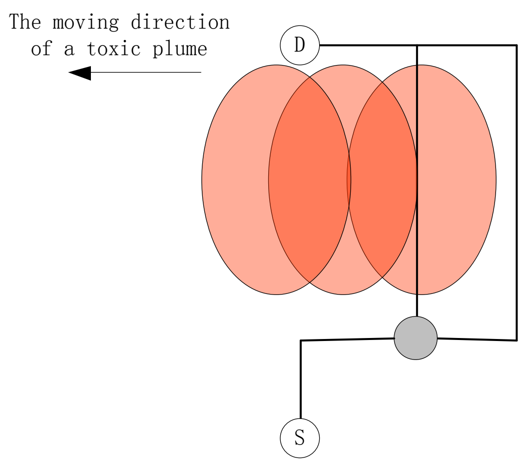



Figure 6.4: An example of the waiting option for the responder

Following the above discussions, we present our second extended A* algorithm, MOAAstar–II/Waiting, which introduces the waiting options to avoid moving obstacles. The data model, CDM–A/Predictions, is still used to structure the spatio-temporal data of the road network affected by plumes. But the affected_time_list associated with each node and each edge is transformed to a sequence of open intervals. Our algorithm is inspired by the path planners developed by Phillips and Likhachev (2011b) and Narayanan et al. (2012) who use safe intervals to represent the time dimension of the search space, but it differs in expansion of the state. Specifically, we also take into account the open intervals of edges and the vehicle speed in generation of successors of the state.

**MOAAstar–II/Waiting**

```
1: Initialize source S, destination D, speed, departureTime
2: openSet := ∅, closedSet := ∅
3: generate source state s_source
4: insert s_source in openSet
5: while openSet is not empty do
6:    s := the state in openSet having the lowest f value
7:    if node(s) = D then
8:       return the path from S to D
9:    end if
10:   remove s from openSet
11:   insert s to closedSet
12:   updateOpenSet(s)
13: end while
14: return no-path
```

Figure 6.5: The main structure of MOAAstar–II/Waiting

When the path planner is initialized, the affected_time_list attached to the each road segment is converted to a sequence of temporal intervals, $P_{xy}$=($p_1^{xy}$, $\ldots$, $p_j^{xy}$, $\ldots$, $p_J^{xy}$), $xy \in E$, to represent the available state of road segments. $p_j^{xy} = (t_{oj}^{xy}, t_{cj}^{xy}), t_{oj}^{xy} < t_{cj}^{xy}$, indicates the time period in which the edge is open, where $t_{oj}^{xy}$ is the start time of the $j$-th open interval, $t_{cj}^{xy}$ denotes the end time of the $j$-th open interval, and $J$ is the total number of open intervals of edge $xy$. For example, an affected_time_list with $([0,4],[5,9])$ is converted to a list of open intervals $([4,5],[9,+\infty])$. Similarly, the affected_time_list associated with each node $x$ is also transformed to a sequence of open intervals, $P_x = (p_1^x, ..., p_k^x, ..., p_K^x)$, $x \in N$ and $K$ is the number of open intervals of node $x$. $p_k^x = (t_{ok}^x, t_{ck}^x)$ is the $k$-th open interval of node $x$, $t_{ok}^x$ is the start time of $p_k^x$, and $t_{ck}^x$ denotes the end time of $p_k^x$. In the path planner, we use the open intervals to construct the state space. The state is defined by a pair of the node and the open interval associated with this node, i.e., $s_k^x$=$(x, p_k^x)$. Accordingly, for each node $x$, there is a finite set of states, $S_x = (s_1^x, ..., s_k^x, ..., s_K^x)$, where $s_k^x$ corresponds to the open interval $p_k^x$.

In MOAAstar–II, the input parameters are initialized in the same way as in the algorithm MOAAstar–I, but the algorithm uses the defined state to generate obstacle avoiding routes. Figure 6.5 shows the outline of our extended algorithm. As in regular A* search, the algorithm selects the state with the lowest cost for further expansion (line 6 in Figure 6.5). When a state $s$ is expanded, we generate its successors, using function $UpdateOpenSet(s)$ (see Figure 6.6), and insert them into the *OpenSet* for further expansion. We first estimate if the vehicle can pass through the edge within each open interval $[t_{oi}^{xy}, t_{ci}^{xy}]$ of edge $xy$, considering the open interval of node $x$. We derive the earliest start time from state $s$, i.e., *start_time*, and use it to compute the earliest arrival time of node $y$ based on the length of edge $xy$ and the *speed* (line 4-10 in Figure 6.6). If node $y$ can be reached within one of its open intervals, we generate a new state $s'$ for node $y$ (line 12-13 in Figure 6.6). Because the vehicle can take wait action at node $x$ to avoid the obstacles and to pass through the edge safely, in line 14 (Figure 6.6) we introduce $t_w(s, s')$ to store the waiting time needed for moving from one state $s$ to another state $s'$. The waiting time $t_w(s, s')$ is calculated from the difference between *start_time* from state $s$, and the arrival time at state $s$, i.e., $g(s)$. We compute the heuristic $h(s')$ based on Euclidean distance between node $y$ and the destination $D$ and the *speed*, and use it to update the state $s'$ together with earliest *arrival_time*. Then the updated state $s'$ is inserted into *openSet* for further expansion.

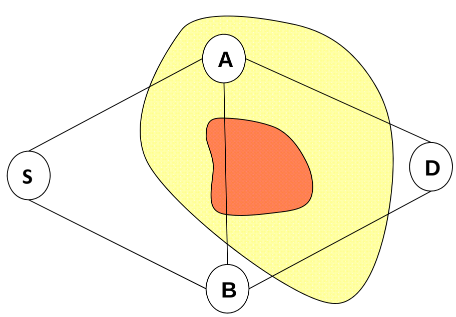
**updateOpenSet($s$)**

1: x := the node associated with $s$<br>
2: $p_k^x := [t_{ok}^x, t_{ck}^x]$, the open interval of $x$ corresponding to $s$<br>
3: **for** each neighbor $y$ of $x$ **do**<br>
4:   **for** each interval $i$ of edge $xy$ **do**<br>
5:     **if** $(t_{ck}^x < t_{oi}^{xy})$ or $(t_{ci}^{xy} < g(s))$ **then**<br>
6:       continue<br>
7:     **else**<br>
8:       start_time := $max(g(s), t_{oi}^{xy})$<br>
9:       arrival_time := start_time + $l_{xy}$/speed<br>
10:       **if** arrival_time $< t_{ci}^{xy}$ **then**<br>
11:         **for** each interval $j$ of node $y$ **do**<br>
12:           **if** ($t_{oj}^y <$ arrival_time) and (arrival_time $< t_{cj}^y$) **then**<br>
13:             generate new state $s' := (y, p_j)$<br>
14:             $t_w(s, s')$ := start_time - $g(s)$<br>
15:           **end if**<br>
16:         **end for**<br>
17:         **if** $s'$ is in closedSet **then**<br>
18:           continue<br>
19:         **end if**<br>
20:         **if** $s'$ is in openSet **then**<br>
21:           **if** arrival_time $< g(s')$ **then**<br>
22:             $g(s')$ := arrival_time<br>
23:             $f(s') := g(s') + h(s')$<br>
24:           **end if**<br>
25:         **else**<br>
26:           $g(s')$ := arrival_time<br>
27:           $f(s') := g(s') + h(s')$<br>
28:           insert $s'$ into openSet<br>
29:         **end if**<br>
30:       **end if**<br>
31:     **end if**<br>
32:   **end for**<br>
33: **end for**

Figure 6.6: UpdateOpenSet($s$)

## 6.4 Algorithm MOAAstar–III/Uncertainty for one-to-one path planning

As already mentioned in Section 2.3.3, the uncertainty in the hazard simulation results is an important issue, and should be considered in the routing among the moving obstacles. In Section 4.4.3, we have presented data model, CDM–C/Uncertainty, which can handle the uncertain moving obstacles. In this section, we present our algorithm, MOAAstar–III/Uncertainty, which can use the information of the uncertainties to compute the safe routes for the responders. The problem we intend to address is as follows. Given a road network that is affected by some smoke plumes, a response unit has to move from a source to a destination in this road network, and departure from a given start time. Here we make the following assumptions: 1). We are given a set of simulation results that are generated from MC simulations of the plume model; 2). The responders are allowed to pass through some moving obstacles if the risk of the moving obstacles they encounter is below a certain level. But they have a limited amount of time for moving under these obstacles, depending on the protective equipment they use. An example where the second assumption holds would be navigation for responders who use the gas mask, as shown in Figure 6.7. As they move through the plumes, their movement is the limited by the amount of oxygen that can be used for a certain period of time. Under the above assumptions, the problem is to compute a feasible route to the destination, taking into account the *risk level* of road segments and junctions as well as the time of passing through the obstacles.




(a) A simple road network affected by smoke plumes with different concentrations of toxic materials. The risk level of the yellow zone is $L2$ ($\geq$ 0.5 parts per million); The risk level of the red zone is $L4$ ($\geq$ 20 parts per million)

(b) Responders wearing gas masks. We assume that the responders have to go from $S$ to $D$, and they are allowed to pass through the yellow zone, but should avoid the red zone (from http://gravv3s.tumblr.com/image/104099515547)

Figure 6.7: Navigation for first responders with gas masks

In MOAAstar–III, we adapt our previously developed algorithm MOAAstar–II for the problem of path planning under uncertain moving obstacles. Extending the concept of safe intervals used by Phillips and Likhachev (2011b) and Narayanan et al. (2012), in MOAAstar–III we distinguish three types of intervals: *full-safe*, *partial-safe* and *non-safe intervals*. These intervals are defined by users based on the *risk level* that we introduced in model CDM–C to describe the status of road segments and junctions. Moreover, in the algorithm we introduce a continuous risk function that accumulates the time of moving along the *partial-safe* roads, and add it into the state as a new state variable (represented by $r$), modeling the risk value as an additional dimension. In addition to that, we introduce a new parameter, $r_{max}$, which indicates the maximal amount time that the responders are allowed to pass through *partial-safe* roads. The objective of algorithm is to try to minimize the arrival time to the destination, constraining the time of passing through *partial-safe* roads (i.e., $r \leq r_{max}$) and avoiding the *non-safe* roads. Because of the addition of the risk function, the search space is also increased, which incurs more computational cost. For addressing this problem, a special technique is required in the algorithm to limit the search area.

### *1. Defining safe intervals*

In our approach, the responders are involved in defining the safe intervals for the search space. In the data model CDM–C/Uncertainty (presented in Section 4.4.3), the uncertainty of components in the road network is now described by the risk_level in RiskTimePeriod. Based on the risk_level, RiskTimePeriod are categorized into the three types of intervals that are mentioned earlier: *full-safe intervals* which are free to pass through, and *partial-safe intervals* that can still be passed through, but have some risks and *non-safe* intervals which are not allowed to pass through. Accordingly, the corresponding *risk levels* are called *full-safe levels*, *partial-safe levels*, and *non-safe levels* respectively. Considering that responders may have different protective equipment, the categorization is left to the responders who classify RiskTimePeriods into the three defined types of intervals. Based on different user profiles, each RiskTimePeriod would fall into a different type of intervals.

More formally, the list of RiskTimePeriods, which is stored in the attribute timeline for each road segment and junction, is divided into the following three temporal sets according to the user profile: $(p_1^\alpha, \ldots, p_q^\alpha, \ldots, p_Q^\alpha)$, $(p_1^\beta, \ldots, p_m^\beta, \ldots, p_M^\beta)$, $(p_1^\gamma, \ldots, p_u^\gamma, \ldots, p_U^\gamma)$, where $p_q^\alpha$ represents the interval that is fully safe; $p_m^\beta$ is the interval that is partially safe; $p_u^\gamma$ indicates the interval that is not safe. $p_q^\alpha = (t_{oq}^\alpha, t_{cq}^\alpha)$, $t_{oq}^\alpha < t_{cq}^\alpha$, $t_{oq}^\alpha$ is the start time of the full-safe interval $p_q^\alpha$, $t_{cq}^\alpha$ denotes the end time of $p^{\alpha}_{q}$, and $Q$ is the total number of the *full-safe intervals*. Similar definitions hold for $p^{\beta}_{m}$ and $p^{\gamma}_{u}$. By grouping together the *partial-safe intervals* and *full-safe intervals*, we can obtain the open intervals, $(p_1, \ldots, p_k, \ldots, p_K)$, as defined in algorithm MOAAstar–II. For example, for a user who defines *risk level* as follows: *full-safe levels*=$\{L1, L2\}$, *partial-safe levels*=$\{L3, L4\}$, and *non-safe levels* = $\{L5\}$, a timeline, as shown in Figure 6.8, can be converted into a list of *full-safe intervals* $([0, 9], [20, +\infty])$, a list of *partial-safe intervals* $([13, 20])$, and a list of open intervals $([0, 9], [13, +\infty])$. It should be noted that we do not consider the *non-safe* intervals in the routing, because they are outside the search space.

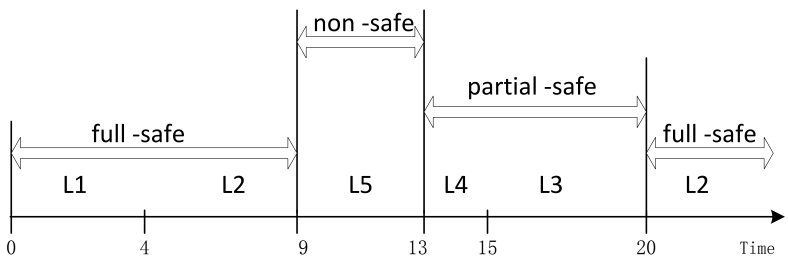



Figure 6.8: A timeline consisting of three types of intervals defined based on the risk levels

*2. Defining state and state dominance*

Like algorithm MOAAstar–II, we still use the open intervals to define the state for building the search space. However, to support path planning with risk function that accumulates the time of moving in the *partial-safe intervals*, we explicitly include the risk function as part of the state space. Formally, a state is defined as follows: $s = (x, p', p'', r)$, where $x$ is the node associated with the state; $p'$ corresponds to one of the open intervals of node $x$, i.e., $p_k$; $p''$ is either a *full-safe interval* $p^{\alpha}_{q}$ or a *partial-safe interval* $p^{\beta}_{m}$; $r$ is a risk variable representing the accumulated time of passing through the *partial-safe interval* of roads. In this way, we can easily see how much time the responder has been going through the *partial-safe* obstacles, given a state $s$. Based on the newly defined state, we build a new search space which has two more dimensions than MOAAstar–II. As risk value $r$ is obtained from a continuous function of time, for a given node $x$ within intervals $p'$ and $p''$, there can be multiples states, all corresponding to different risk values.

Because the dimensionality of our search problem is increased by addition of the continuous risk function into the state, the search space is also enlarged, which results in more computing time and space needed for the algorithm to search for feasible paths. In this study, a state dominance relationship is defined and applied to reduce the search space. State dominance has been used in many planning problems to limit the search space, and thus to improve the computation speed (Gonzalez and Stentz, 2005; Gonzalez et al., 2012a). The idea of state dominance is that if a state $s$ is dominated by another state $s'$, the solution obtained from $s$ can not be better than the solution found from $s'$, which means that state $s$ would unlikely contribute to the optimal solution. Using the state dominance, we can facilitate the search process by identifying the dominated states and pruning them from further expansions in the search, without compromising the optimality of the solution.

For our path planning with the risk function, we derive a state relationship based on the state defined above. In the algorithm, we aim to minimize the travel cost for the responders, while limiting the total risk that is accumulated in passing through *partial-safe* roads. In this sense, a state with less travel cost and lower risk would dominate the one with more travel cost and higher risk. More specifically, given two states $s$ and $s'$, which refer to the same node within the same intervals, but differ on the risk values: $s_u = \{x, p', p'', r_u\}$, $s_v = \{x, p', p'', r_v\}$, if ($g(s_u) <= g(s_v)$ and $r_u < r_v$) or ($g(s_u) < g(s_v)$ and $r_u <= r_v$), then we say that state $s_u$ dominates state $s_v$. For state $s$, $g(s)$ represents the least cost found so far from the source to $s$; $r$ denotes the amount of risk that has been accumulated from the source to $s$.

*3. Discretizing the search space*

When the vehicle moves along an edge that has multiple *full and partial safe intervals*, the accumulated risk varies with the starting time from one node of the edge to the other. For example, as shown in Figure 6.9, the vehicle can departure from the earliest start time $t_0$, and moves at a constant *speed*. The dash lines have the equal length of time and represent the time periods that the vehicle needs to travel through the edge. Different waiting times, $t_{aw_1}$, $t_{aw_2}$, $t_{aw_3}$ , can be introduced and allow the vehicle to start its movement at different times. The addition of risk $\Delta r$ is calculated as the sum of the time periods when the *partial-safe intervals* overlap the traveling period. It is easy to see that $\Delta r$ is a value between $\Delta r^{min} = 0$ and $\Delta r^{max} = l/speed$. Figure 6.10 shows the accumulated risk values that correspond to the waiting times. As we can see, given different waiting times, the accumulated risk values are also different. Although waiting would make the responder take more time to arrive the destination, it may have some advantages in limiting the risks that the responders would encounter, and should be considered in the routing.

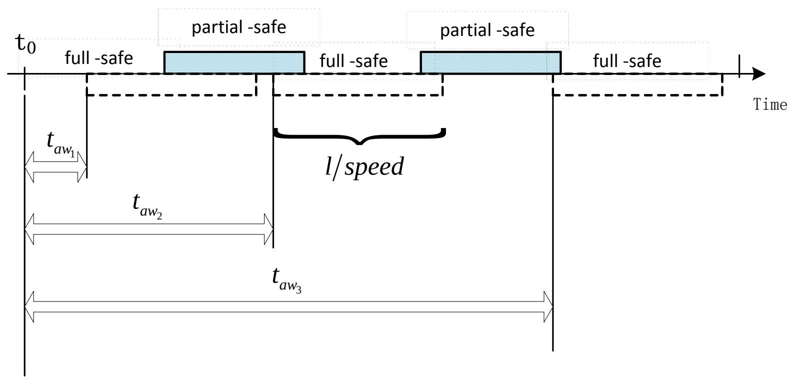



Figure 6.9: An example of different waiting times in passing through an edge

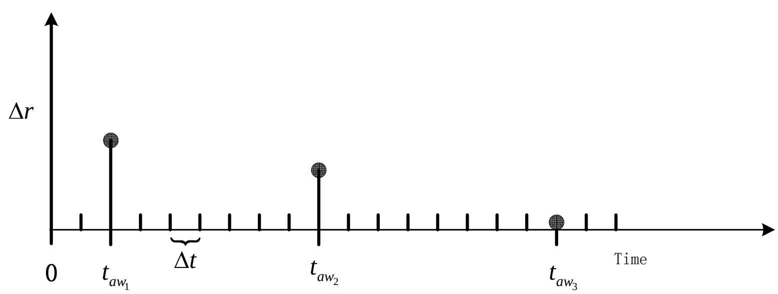



Figure 6.10: The addition of risk with different waiting times

Following the above discussions, we introduce an additional waiting option that would lead to less risk, plus the waiting until the edge is safe. Because using contiguous timeseries to generate states would cause the size of the search space significantly enlarged during the search, inspired by the approach used in Van Den Berg and Overmars (2005), in the algorithm we discretize the waiting time series into small timesteps, $\Delta t$, to generate succeeding states for further expansions, as shown in Figure 6.10. We assume that the chosen time step $\Delta t$ is small enough to be able to generate all possible states. Special constraints will be imposed on the waiting time, limiting the generation of successors.

*4. Planning in the defined search space*

Figure 6.11 shows the main structure of our third A* extended algorithm, MOAAstar–III/Uncertainty. Similar to algorithm MOAAstar–II, the algorithm starts with a given source and destination, speed and the departure time. When the state $s$ with the lowest cost is selected for further expansion, On line 13 (Figure 6.11) we use function GenerateSuccessors $(s)$ (Figure 6.12) to find successors of state $s$, considering all possible *full and partial-safe intervals* in transition from state $s$. On line 15 (see Figure 6.11), using the state dominance function defined in Figure 6.13, we iterate over all states in the successors to check if the successor is dominated by the states that have been found. We only insert the non-dominated successors into the open set for future expansion, which would reduce the search space for the path planner.

**MOAAstar–III/Uncertainty**

```
1: Initialize source S, destination D
2: initialize r_max, speed, departureTime
3: openSet = ∅, closedSet = ∅
4: generate source state s_source
5: insert s_source in openSet
6: while openSet is not empty do
7:    s = the state in openSet having the lowest f value
8:    if node(s) = D then
9:       return the path from source S to destination D
10:   end if
11:   remove s from openSet
12:   insert s to closedSet
13:   successors = generateSuccessors(s)
14:   for each successor s' in successors do
15:      if ¬ check_dominated(s') then
16:         insert s' into openSet
17:      end if
18:   end for
19: end while
20: return no-path
```

Figure 6.11: The main structure of MOAAstar–III/Uncertainty

In generation of successors (Figure 6.12), the algorithm first examines each open interval $[t_{ci}^{xy}, t_{ci}^{xy}]$ of edge $xy$ to see if it overlaps with $[g(s), t_{ck}^{x}]$ (line 5-7, Figure 6.12). Only the overlapped intervals are safe for the vehicle to pass through the edge, and will be considered. Then we use an incremental approach to generate possible successors, increasing the waiting time with the fixed time step. To derive the maximum waiting time, $t_{aw}^{max}$, that the vehicle can wait after the earliest start time, we distinguish *partial-safe interval* and *full-safe interval* for state $s$ (line 9-14, Figure 6.12): 1). If $s$ is within a *full-safe interval*, $t_{aw}^{max}$ is set to the difference between the closed time of *full-safe interval* $t_{cq}^{x,\alpha}$ and the earliest start time; 2). If $s$ is within a *partial-safe interval*, $t_{aw}^{max}$ is set

**generateSuccessors($s$)**

```
s = (x, p', p'', r), x = node(s), p' = [t^x_{ok}, t^x_{ck}]
new_states = ∅
for each neighbor y of x do
  for each open interval i of edge xy do
    if t^x_{ck} < t^{xy}_{oi} or t^{xy}_{ci} < g(s) then
      continue
    end if
    earliest_start_time = max(g(s), t^{xy}_{oi})
    if s is in a full safe interval q, p^{x,α}_q then
      t^{max}_{aw} = t^{x,α}_{cq} - earliest_start_time
    end if
    if s is in a partial safe interval then
      t^{max}_{aw} = 0
    end if
    t_{aw} = 0
    while t_{aw} <= t^{max}_{aw} do
      start_time = earliest_start_time + t_{aw}
      if start_time > t^{xy}_{ci} then
        break;
      end if
      arrival_time = start_time + l_{xy}/speed
      calculate the incremental risk Δr
      if arrival_time < t^{xy}_{ci} and r(s) + Δr <= r_{max} then
        generate state s' of node y
        t_w(s, s') = start_time - g(s)
        if s' is in closedSet then
          continue
        end if
        if s' is in openSet then
          if arrival_time < g(s') then
            g(s') = arrival_time
            f(s') = g(s') + h(s')
          end if
        else
          g(s') = arrival_time
          f(s') = g(s') + h(s')
          insert s' into new_states
        end if
      end if
      if ¬ check_increment(start_time, xy) then
        break;
      end if
      t_{aw} = t_{aw} + Δt
    end while
  end for
end for
return new_states
```

Figure 6.12: generateSuccessors($s$)

**Check_dominated($s$)**

$s = (x, p', p'', r)$, $x = node(s)$
**for** each $\hat{s}$ in openSet $\cup$ closedSet **do**
  **if** $node(\hat{s}) = node(s)$ and $p'(\hat{s}) = p'(s)$ and $p''(\hat{s}) = p''(s)$ **then**
    **if** $g(s) > g(\hat{s})$ and $r(s) >= r(\hat{s})$ **then**
      return true
    **end if**
    **if** $g(s) >= g(\hat{s})$ and $r(s) > r(\hat{s})$ **then**
      return true
    **end if**
  **end if**
  return false
**end for**

Figure 6.13: Check_dominated($s$)

0. This is because that waiting within the *partial-safe interval* does not provide the possibility of reducing the risk value of the successors, and would not be advantageous. The obtained $t_{aw}^{max}$ is used as a limit on the increment of waiting time $t_{aw}$ on line 16.

**Check_increment(start_time, xy)**

$p_Q^{xy,\alpha} = [t_{cQ}^{xy,\alpha}, t_{cQ}^{xy,\alpha}]$, the last full-safe interval of edge $xy$
$p_M^{xy,\beta} = [t_{cM}^{xy,\beta}, t_{cM}^{xy,\beta}]$, the last partial-safe interval of edge $xy$
**if** start_time $> t_{oQ}^{xy,\alpha}$ and $t_{cQ}^{xy,\alpha} = inf$ **then**
  return false
**end if**
**if** start_time $> t_{oM}^{xy,\beta}$ and $t_{cM}^{xy,\beta} = inf$ **then**
  return false
**end if**
return true

Figure 6.14: Check_increment(start_time, xy)

In every loop, on line 18 (Figure 6.12) we first check the given waiting time to guarantee that the vehicle starts before the edge is closed. After that, we calculate the possible arrival time from node $x$ to $y$ (line 21, Figure 6.12), and estimate the risk increment, $\Delta r$, that would be accumulated during the travel through edge $xy$ (line 22, Figure 6.12). If the vehicle can safely pass through the edge and arrives with a new risk lower than $r_{max}$ (line 23, Figure 6.12), we generate a successor $s'$, and computes the waiting time needed for transition from $s$ to the state $s'$ (line 24-25, Figure 6.12). As in the regular A* search, we ignore the state $s'$ that is in the *closedset*, because it has been expanded in previous search (line 26-28, Figure 6.12). If the state is in the *openSet* and the newly found path has a shorter time, we update the existing state with the new arrival time and the estimated traveling time $h(s')$ (line 29-33, Figure 6.12). The heuristic $h(s')$ is calculated based on Euclidean distance between node $y$ and the destination $D$ and the *speed*. The newly created state is inserted into the *openSet* for further expansion (line 35-37, Figure 6.12). It is important to note that the *full-safe interval* of state $s$, $t_{cq}^{x,\alpha}$, and the closed time of edge $xy$, $t_{ci}^{xy}$ could be infinity, which would cause an infinite loop in the algorithm. To prevent this, we adopt an additional function (Figure 6.14) to check if the increment of waiting time should be stopped, making the algorithm generate a finite set of successors. In Figure 6.9 and Figure 6.10, we have shown that given an earliest start time and a traveling period, the accumulated risk changes with the waiting time. In the following, we will prove that during the increment of the waiting time $t_{aw}$, the generated states that do not meet the conditions on line 40 (Figure 6.12) will be dominated by a state generated earlier.

**Theorem 2.** *If state $s$ is within a full-safe interval that has an infinite end time, $t_{cq}^{x,\alpha} = inf$, and the considered open interval on edge $xy$ also has an infinite end time, $t_{ci}^{xy} = inf$, there exists a waiting time $t_{aw}^0$, such that the states that are generated after $t_{aw}^0$ will be dominated by the state that corresponds to $t_{aw}^0$.*

*Proof.* $t_{ci}^{xy} = inf$ implies that either the end time of the last full-safe interval, $t_{cQ}^{xy,\alpha}$, is infinity or the end time of the last partial-safe interval, $t_{cM}^{xy,\beta}$, is infinity. In the case that $t_{cQ}^{xy,\alpha} = inf$, we select the start time of the last full-safe interval as the start time of the vehicle, then get $\Delta r = 0$. We denote the corresponding waiting time by $t_{aw}^0$. For any $t_{aw} > t_{aw}^0$, the generated state $s$ from $t_{aw}$ would have a longer path, thus $g(s) > g(s_0)$ and $r(s) = r(s_0) = 0$. Based on our defined state dominance relationship, state $s$ is dominated by $s_0$. Similarly, in the case that $t_{cM}^{xy,\beta} = inf$, we select the start time of the last partial-safe interval as the start time of the vehicle, and also represent the waiting time by $t_{aw}^0$. The obtained $\Delta r$ is $\Delta r^{max}$. For any $t_{aw} > t_{aw}^0$, the generated state $s$ from $t_{aw}$ would have a larger path cost, thus $g(s) > g(s_0)$ and $r(s) = r(s_0) = \Delta r^{max}$. $s$ is also dominated by $s_0$.

## 6.5 The algorithm for one-to-many path planning

The one-to-many path planning problem can be seen as a variant of the traveling salesman problem (TSP). Using simple but effective rules, insertion heuristics have been applied to many variants of TSP, and can produce quality solutions (close to optimal) in a fast way. Therefore, in this study we use the insertion heuristic to find near optimal solutions to the one-to-many path planning problems with moving obstacles.

Figure 6.15 describes our one-to-many path planning algorithm based on the insertion method. The objective of the algorithm is to minimize the total trip duration, which includes the total traveling time and the operation time of tasks. Given a set of $k$ tasks, $T = \{T_1, \ldots, T_i, \ldots, T_k\}$, $1 <= i <= k$, that need to be allocated to a responder. Each task corresponds to an instance of class Task in the data model, CDM–B/Multi-destination. The algorithm calculates the path cost of insertion of a evaluated task $T_i$ in each position. Then it selects the position that makes the cost of the new path be minimal. Finally it chooses the task that has the maximum path cost to ensure that worst task is inserted into the path $P$ first. Each trip between two tasks in the path is calculated by the modified A* algorithms for one-to-one path planning, based on the given speed as well as the departure time of vehicles.

Because responders should spend a certain amount of time in performing their tasks, the operation time of each task is also important in the path planning among moving obstacles. In our system, the operation time is extracted from the operation_time of TaskByVehicle in the logical data model 4.5, and added into the algorithm to derive the time of travel to the next task as follows:

$$t_{\mathrm{D}}(T_j) = t_{\mathrm{A}}(T_j) + t_{\mathrm{O}}(T_j) \tag{6.1}$$

Here $t_{\mathrm{D}}(T_j)$ is the departure time from the position of task $T_j$, $t_{\mathrm{A}}(T_j)$ is the arrival time to the position of task $T_j$, and $t_{\mathrm{O}}(T_j)$ is the operation time of task $T_j$.

In many crisis situations, emergency tasks require a specific time for arrivals and responders have to reach destinations in a limited amount of time. One the other hand, the influence of moving obstacles on roads can cause longer trips of vehicles, and result in delays beyond the required time for responders. Therefore, we use the completion time of tasks as one of criteria in selecting the routes that responders should follow, taking into account the required arrival time of tasks and the delays caused by moving obstacles. In this study, we provide a multi-objective function that tries to strike a balance between minimizing the number of delays, the sum of all delays, and the task-completion time: Let $b_{\mathrm{delay}}(T_j)$ be a binary variable to indicate if the vehicle arrives later than the required arrival time of task $T_j$, and is defined as follows:

$$b_{\mathrm{delay}}(T_j) = \begin{cases} 1 & \text{if } t_{\mathrm{A}}(T_j) > t_{\mathrm{RA}}(T_j) \\ 0 & \text{if } t_{\mathrm{A}}(T_j) \leq t_{\mathrm{RA}}(T_j) \end{cases} \tag{6.2}$$

**Incremental insertion method**

```
source, departure_time, speed
T = {T_1, T_2, ··· , T_k}
P = ∅ ; // the sequence of tasks forming the path
while T ≠ ∅ do
   for i = 1 → |T| do
      Select task T_i from T
      if (|P|= 0) then
         Calculate the path from source to T_i
      else
         for j = 1 → |P| do
            Insert task T_i in the position j of P
            Calculate the path from source to T_{I_1}
            for l = 2 → |P| + 1 do
               Calculate the departure time from
                     the position of T_{I_{l-1}} by formula (1)
               Calculate the path from T_{I_{l-1}} to T_{I_l}
            end for
         end for
      end if
      Insert T_i in the position that produces the minimum path cost
   end for
   Select T_max with the maximum path cost
   P = P ∪ {T_max}
   T = T\{T_max}
end while
```

Figure 6.15: Incremental insertion method (adapted from Skiena (2008))

where $t_{\text{A}}(T_j)$ is the actual arrival time, and $t_{\text{RA}}(T_j)$ is the required arrival time of task $j$. Let $t_{\text{delay}}(T_j)$ denotes the time delay between $t_{\text{A}}(T_j)$ and $t_{\text{RA}}(T_j)$ of task $j$, then it can be written as follows:

$$t_{\text{delay}}(T_j) = \begin{cases} t_{\text{A}}(T_j) - t_{\text{RA}}(T_j) & \text{if } t_{\text{A}}(T_j) > t_{\text{RA}}(T_j) \\ 0 & \text{if } t_{\text{A}}(T_j) \leq t_{\text{RA}}(T_j) \end{cases} \tag{6.3}$$

Using above notations, we formulate the following multi-objective function for minimization of path cost of a vehicle:

$$F(v, P) = CT(v, P) + w_1 \times DN(v, P) + w_2 \times DT(v, P) \tag{6.4}$$

where $F(v, P)$ is the cost of the entire path $P$ of vehicle $v$, $CT(v, P)$ is the completion time of all tasks in path $P$ (in minutes), $DN(v, P)$ represents the number of delays, $\sum_{j=1}^{k} b_{\text{delay}}(T_j)$, $DT(v, P)$ corresponds to the sum of all time delays (in minutes), $\sum_{j=1}^{k} t_{\text{delay}}(T_j)$, and $w_1$, $w_2$ are penalty weights.

The weights, $w_1$ and $w_2$, are set based on the priority of the corresponding objective in the function. Specifically, an objective with higher priority has a higher weight. In our research, we first try to minimize the number of delays first $DN(v, P)$, next minimize the sum of all delays $DT(v, P)$, and then the completion time of tasks $CT(v, P)$. Given these priorities, we set the penalty weights in the following way: $w_1 > w_2 > 1$.

## 6.6 The algorithm for many-to-many path planning

Considering computational efficiency in the multi-agent coordination (Schoenig and Pagnucco, 2011), in this study we apply Sequential Single-Item (SSI) (Koenig et al., 2006) for allocation of tasks to vehicles.

**Sequential single-item (SSI) auction**

```
T = {T_1, T_2, ··· , T_k}
T(v) = ∅
while T ≠ ∅ do
    for each task T_i ∈ T, 1 <= i <= |T| do
        for each vehicle agent v ∈ V do
            estimate the cost c of carrying out tasks T(v) ∪ {T_i}
            send bid(c, T_i) to the task allocation agent
        end for
    end for
    the task allocation agent
        selects the v_s with the smallest bid (c_s, T_s)
    T = T\{T_s}
    T(v_s) = T(v_s) ∪ {T_s}
end while
```

Figure 6.16: Sequential Single-Item (SSI) auction (adapted from Koenig et al. (2006))

Figure 6.16 shows the main structure of SSI auctions. During auctions, the Task Allocation Agent first makes an announcement to all Vehicle Agents notifying them about new tasks. Then the Vehicle Agents estimate the cost of completing the unallocated tasks and the already assigned tasks, using the one-to-many path algorithm presented in Section 6.5, and submit bids with the calculated cost. We use the minmax team objective (Lagoudakis et al., 2005) to minimize the maximum path cost over all vehicles, i.e., $\min \max_{i=1}^{|V|} F(v_i, P_i)$. For this objective, each Vehicle Agent should bid with its cost $c = F(v_i, P_i)$ on the target task. Finally the Task Allocation Agent collects the bids and determines the winner of the auction, assigning a single task to the Vehicle Agent with the lowest cost. The above described process is repeated until all tasks are assigned.

In our research, we use a dynamic method for re-allocation of tasks. Because the predictions from hazard simulation change with real sensor measurements, the previously generated allocation plans can not be applied to the new situations and need to be adapted. The method repeats auctions while tasks are being executed. When new predicted information of hazards is gathered, the task allocation is restarted and all unaccomplished tasks will be auctioned again. This enables the system to dynamically allocate tasks in the environment affected disasters, providing the ability to adapt to changing conditions.

## 6.7 Concluding remarks

In this chapter, we have provided a suite of algorithms for different types of path planning problems, aiming to address the following research question:

(4) What algorithms should be developed for path planning among moving obstacles?

For one-to-one path planning, we modify the A* algorithm for different situations, incorporating not only the predicted information of the road network affected by moving obstacles, but also the information of responders, such as the speed of vehicles, departure time, the capability of passing through the obstacles. With respect to one-to-many path planning, we use insertion heuristics and provide a multi-objective function to obtain a balance between the number of delays, the sum of all delays, and the task-completion time. Regarding the many-to-many path planning, we apply an approach based on SSI auctions, in which two types of agents, i.e., the Task Allocation Agent and the Vehicle Agents, are involved in determining task allocation via bidding processes.

# 7 IMPLEMENTATIONS AND CASE STUDIES

In the previous chapters, we have elaborated on the design and development of agents (definition of roles, functionalities, interactions etc.), routing algorithms, and spatial-temporal data models that structure the data required for routing and allocation process. This chapter will come to the testing of these developments and evaluation of our navigation system. In Section 7.1, we give the details of the software tools used for the implementation of our system. We have applied the prototype system to the case of a simulated fire event, comparing different aspects between our algorithm and the classical algorithm. A fire simulation model was used to generate datasets about the spread of the fire. The experimental results are shown in Section 7.2. In Section 7.3, we apply the developed navigation system to the four considered navigation cases that involve one/multiple responders and one/multiple destinations. For situations with uncertain moving obstacles, we test the functionality and performance of our algorithm under different setting of parameters, using artificial datasets (see Section 7.4). This chapter is closed with some discussions in Section 7.5.

## 7.1 Implementation

Using the data models in Chapter 4, the structure of the system presented in Chapter 5, and the algorithms in Chapter 6, a prototype of multi-agent based navigation system has been implemented to support one or multiple first responders among moving obstacles, integrating agent technologies, geo-database management systems, and GIS tools. We also developed a server which can handle the one-to-one navigation request from the client application, calculate routes avoiding moving obstacles, and deliver the route results to the client application via the Internet (see Appendix A).

The distributed multi-agent system is developed based on the JADE (Java Agent Development Framework). JADE provides communication service between agents on different machines, and defines a collection of pre-defined behaviours to simply the development of agents (Bellifemine et al., 2005). It has been widely applied in both academic and industrial areas (Filippoupolitis et al., 2009; Rothkrantz, 2009; Filippoupolitis and Gelenbe, 2009). In our system, we use JADE as underlying agent infrastructure and combine it with another agent-based toolkit, GeoMASON (Sullivan et al., 2010). GeoMASON is a GIS extension of simulation toolkit Mason (Luke et al., 2004), using functions provided by two Java GIS software libraries, GeoTools (http://geotools.org/) and Java Topology Suite (JTS, http://www.vividsolutions.com/jts/jtshome.htm). The combination allows agents to perform spatial data processing, manipulation, and analysis using the utility methods provided by GeoMASON, enhancing the JADE agents with GIS functionalities.

All the needed data and calculated results are structured and stored in the geo-DBMS. The proposed spatial data model is realized in the object-relational database PostgreSQL with PostGIS extension (www.postgis.org). We use the spatial data types (such as Point, LineString, Polygon, etc.) provided by PostGIS to deal with GIS data. The calculated routes, together with the data about moving obstacles and relief vehicles, are delivered to GeoMason to configure the visualization module, and are displayed to users through both 2D and 3D viewers, enabling them to gain accurate impressions of the current and future situations. In 2D visualization, we use OpenStreetMap as a base map to provide an overview of planned paths. The 3D viewer is built on top of an open source visualization tool, OSM2World (`www.osm2world.org`) that builds three dimensional models of the environment from OpenStreetMap data. Through the construction of the 3D visualization, situational awareness is enhanced by providing information on the surroundings, such as houses, gardens, etc., that might not initially be included in the street network model.

## 7.2 Application to the case of forest fires

Our prototype system has also been applied to the case of a simulated fire event in San Sebastián, Spain. We use a fire simulation model to calculate the fire evolution, and simulate several scenarios in which one or more fires take place in a forest located in the eastern part of the city. The considered area includes a road network consisting of 1717 edges and 1661 nodes. The fire simulator generates the fire spread dataset within the given area in seconds, starting from time t=0 min to time t=20 min. The information regarding the status of the road network is collected and used for instantiating the model CDM–A/Predictions (see Section 4.4.1). Paths between locations are calculated by using both the algorithm MOAAstar–I/Non-waiting (presented in Section 6.2) and the classical A* algorithm, and are visualized in the developed 3D viewer.

### 7.2.1 *Intersection of the fire-affected area with the road network*

For the prediction of forest fires, a cell-based fire simulation model developed by Moreno et al. (2011) is used to generate datasets of fire-affected areas. The fire simulation method divides the topography into a grid of square cells. Each cell contains both static information, such as position, size (i.e., 3 meters), type, and the burning rate depending on its type, and the runtime information, such as the quantity of combustible, the power intensity of the fire, and the state of the fire. The fire simulation system, integrated with passive data from different sources and dynamic events, including real-time changes in the weather conditions, calculates the spread of the forest fire and updates the runtime information of forest cells calculated during each simulation step. By grouping the cells according to the cell state and time step, we create a set of moving polygons that overlap a certain road network. Considering that each cell in the simulation has a certain width, we introduce a new buffer for each road-center line to represent the road network, extract all the road segments and junctions inside affected areas, and store them with their affected time periods in the database.

### 7.2.2 *Route safety*

To evaluate the safety of the route, we provide a method to quantify the safety value of edges and routes. Our method is similar to the one proposed by Shastri (2006) that introduces the margin of safety of nodes, but uses the affected time of edges to evaluate the safety of routes. The safety of each edge is expressed as difference between the time when fires block the edge and the estimated time when the responder arrive at the target node of the edge. Mathematically, the safety of an edge $n_i n_{i+1}$, $S_{n_i n_{i+1}}$, is

$$S_{n_i n_{i+1}} = t_c^{n_i n_{i+1}} - t^{n_{i+1}} \tag{7.1}$$

Here $t_c^{n_i n_{i+1}}$ is the closed time of edge $n_i n_{i+1}$; $t^{n_{i+1}}$ is the estimated time of reaching node $n_{i+1}$ through edge $n_i n_{i+1}$. As illustrated in Figure 7.1, if $S_{n_i n_{i+1}} > 0$, the edge is considered safe; If $S_{n_i n_{i+1}} <= 0$, the edge is considered not safe.

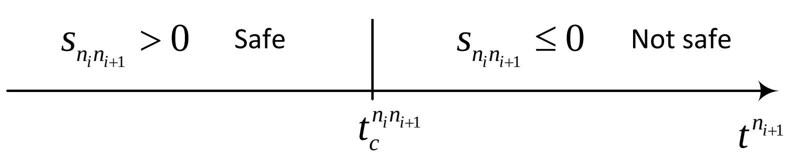



Figure 7.1: the safety of edge $n_i n_{i+1}$

Because the safety of a route mainly depends on the most unsafe edge along the route, the minimum of safety values of edges is selected as the route safety. Let $R = \{n_0, n_1, \ldots, n_k\}$ be one of routes from $S$ to $D$, where $n_0, n_1, \ldots, n_k$ are the nodes along the route, $n_0 = S$, $n_k = D$. The safety of the entire route can be computed by using the following formula (Shastri, 2006):

$$S_R = min(S_{n_0n_1}, S_{n_1n_2}, \ldots, S_{n_{k-1}n_k}) \tag{7.2}$$

Similar to the above definition of edge safety, if $S_R > 0$, the route is considered safe; If $S_R <= 0$, the route is considered not safe. The higher the safety value, the more safe the route is. $+\infty$ means the route is completely safe (i.e., not affected by the predicted fires).

### 7.2.3 *Scenario 1: navigation for one responder to one destination, avoiding one fire-affected area*

Considering that different vehicle types have different maximum moving speeds, we compare relief routes for different speeds to evaluate the practical application of our route planner. Table 7.1 shows the results of our experiments. In the first situation, where the relief vehicle is moving at a speed of 20 km/h, our algorithm and the standard A* algorithm produce different routes, depicted in Figure 7.2. The light blue line is the route calculated by our algorithm, and the brown line represents the shortest path without considering the fire spread. The results indicate that when fires are moving fast and affect the environment rapidly, the vehicle at a speed of 20 km/h can not safely arrive at the destination along the shortest route, because the route could be blocked by fires before the vehicle can pass through. Our algorithm finds a new route that makes the responding unit detour to avoid fires and is safer than the shortest one.

Continuing our analysis, Figure 7.3 depicts another situation in which the shortest path and the calculated route are the same at given speeds of 30 km/h and 50 km/h. As shown in Table 7.1, the vehicle in this situation is moving faster, which leads to a shorter path and less traveling time, compared to the vehicle at a speed of 20 km/h. The Table 7.1 also indicates the vehicle moving at a speed of 50 km/h has a higher safety value than the vehicle at a speed of 30 km/h. By testing different speeds in the application, the emergency manager can determine the minimum speed required to safely pass through the affected region or to follow a specific route.

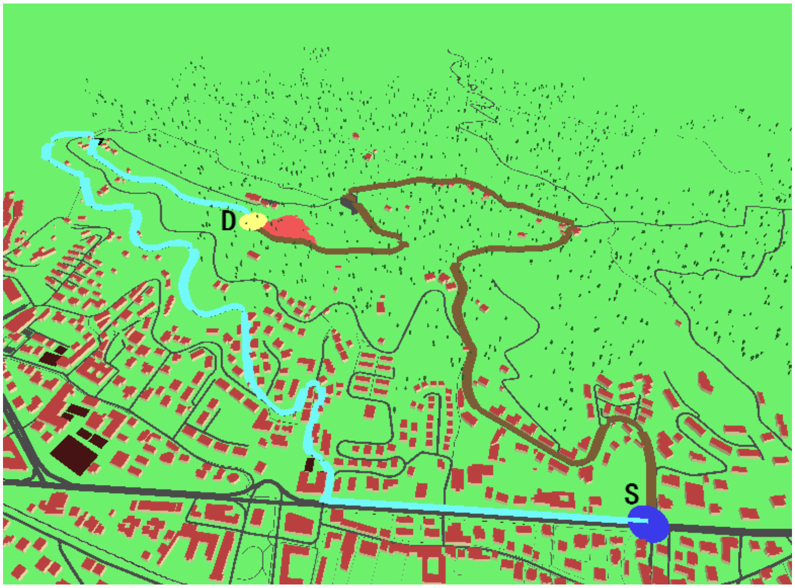



Figure 7.2: The calculated paths (speed=20 km/h) from source S (in blue) to destination D (in yellow) through the environment with one fire-affected area (in red)


Figure 7.3: The calculated paths (speed=30, 50 km/h) from source S (in blue) to destination D (in yellow) through the environment with one fire-affected area (in red)

Table 7.1: Calculated results considering different speeds in fires

| | Route ID | Distance (km) | Total travel time (mins) | Route safety (mins) |
|---|---|---|---|---|
| Speed =20 km/h | R0 | 2.56 | $7.7^{x}$ | $-1.8^{x}$ |
| | R1 | 3.00 | 9.0 | $+\infty$ |
| Speed =30 km/h | R0 | 2.56 | $5.1^{x}$ | $0.7^{x}$ |
| | R2 | 2.56 | 5.1 | 0.7 |
| Speed =50 km/h | R0 | 2.56 | $3.1^{x}$ | $2.7^{x}$ |
| | R3 | 2.56 | 3.1 | 2.7 |

Notes:
[1] The vehicles considered in this scenario departure at time t=0 min
[2] R0: The shortest route calculated by the standard A* algorithm
[3] R1: The route calculated by the algorithm, MOAAstar–I/Non-waiting, given a speed of 20 km/h
[4] R2: The route calculated by the algorithm, MOAAstar–I/Non-waiting, given a speed of 30 km/h (the distance of R2 equals the distance of R0)
[5] R3: The route calculated by the algorithm, MOAAstar–I/Non-waiting, given a speed of 50 km/h (the distance of R3 equals the distance of R0)
[6] $+\infty$: This route is completely safe from t=0 min to t=20 min
[7] x: It is estimated based on the shortest route without considering the moving obstacles

### *7.2.4 Scenario 2: navigation for multiple responders to one destination, avoiding multiple-affected areas*

In this scenario, we study the navigation case that multiple rescue vehicles have to be routed to one destination avoiding multiple fire-affected areas. The considered vehicles have different maximal speeds, and start moving from different locations at different time instants. Our algorithm calculates routes avoiding fires, considering both the speed of vehicles and their departure times. The calculated results are shown in Table 7.2. Because of the fact that the shortest routes could be blocked by the fires, emergency plans made based on estimation of arrival time of the shortest route will not be feasible due to possible delays. As we can see from the table that, although vehicle $v_1$ can arrive at the destination on time, the time difference between arrival time of the shortest route and arrival time of obstacle-avoiding route for vehicle $v_2$ is about 3.5 min, and vehicle $v_3$ has a time difference of 4.5 min. Because responders often work in groups, a reliable estimation of their arrival time at the field site is very important for rapid emergency operations. A lack of consideration of possible delays caused by fires could significantly slow the response process. Figure 7.4 shows a snapshot of routes calculated by our algorithm. The results indicate that our algorithm can not only deal with multiple fire-affected areas, but also give a more reliable estimation of arrival time for different types of vehicles starting from different places and different time instances, which would make emergency plans more effective and contributes to an improvement of performance of the response units.

Table 7.2: Calculated results for 3 vehicles in fires

| Vehicle ID | Route ID | Departure time (min) | Distance (km) | Total travel time (mins) | Arrival time (min) |
|---|---|---|---|---|---|
| $v_1$ | R0 | 2.0 | 3.0 | $6.0^x$ | $8.0^x$ |
| (30 km/h) | R1 | 2.0 | 3.0 | 6.0 | 8.0 |
| $v_2$ | R2 | 5.0 | 1.8 | $5.3^x$ | $10.3^x$ |
| (20 km/h) | R3 | 5.0 | 2.9 | 8.8 | 13.8 |
| $v_3$ | R4 | 8.0 | 2.2 | $6.5^x$ | $14.5^x$ |
| (20 km/h) | R5 | 8.0 | 3.7 | 11.0 | 19.0 |

Notes:
[1] R0, R2, R4: The shortest routes from different sources to the same destination, calculated by the standard A* algorithm
[2] R1: The route calculated by the algorithm, MOAAstar–I/Non-waiting, given a speed of 30 km/h and a departure time t=2.0 min (the route R1 and the shortest route R0 are the same)
[3] R3: The route calculated by the algorithm, MOAAstar–I/Non-waiting, given a speed of 20 km/h and a departure time t=5.0 min
[4] R5: The route calculated by the algorithm, MOAAstar–I/Non-waiting, given a speed of 20 km/h and a departure time t=8.0 min
[5] x: It is estimated based on the shortest route without considering the moving obstacles


Figure 7.4: The calculated paths for three vehicles among multiple fire-affected areas (Vehicle $v_1$ from source S1 (in blue) to destination D (in yellow); Vehicle $v_2$ from source S2 (in purple) to destination D (in yellow); Vehicle $v_3$ from source S3 (in brown) to destination D (in yellow))

## 7.3 Application to the cases involving one/multiple responders one/multiple destinations, and moving obstacles

To address the four navigation cases mentioned in Section 3.7, we have applied the developed data model, CDM–B/Multi-destination (see Section 4.4.2), and used the algorithm, MOAAstar–II/Waiting (see Section 6.3), the one-to-many path planning algorithm (see Section 6.5), and the many-to-many path planning algorithm (see Section 6.6). We tested them with the road network dataset in Delft[1], the Netherlands. The network is composed of 1586 edges and 1780 nodes. We consider the following specific crisis response scenario. Suppose that a poisonous material has been accidentally released into the city and some plumes affect the central part of the city. There are a number of rescue tasks and the first responders are distributed to assist injured people who require medical service, to do measurements and observations, and to get the affected people out of the dangerous areas. To help carry out these tasks, the system calculates the obstacle-avoiding routes for the responders in the following four navigation cases. The calculated results are shown through 2D visualization.

### 7.3.1 *Case 1: $< O, O, S, M >$navigation of one responder to one destination*

In this case, an emergency manager needs to select a vehicle to go to a destination, meeting requirements on the response time. Here we assume that responders should reach the destination in 10 min. The emergency manager first checks the availability of vehicle $v_1$ which can move at a speed of 40

[1] http://3dgeoinfo.bk.tudelft.nl/projects/navigation/

km/h. The system computes the obstacle avoiding routes for this vehicle. For comparison purposes, we also use the standard A* algorithm to calculate the shortest routes. Table 7.3 shows the calculated results and Figure 7.5 displays the calculated routes on the map. The light-gray line is the shortest route, and the dark line is the route calculated by the algorithm MOAAstar–II considering the speed of 40 km/h. As shown in the table, $v_1$ has to wait for a long time to avoid obstacles according to the estimation provided by our algorithm. This results in that vehicle $v_1$ would fail to arrive within the required response time.

Table 7.3: Calculated results of the case $< O, O, S, M >$

| Vehicle ID | Route ID | Distance (km) | Total waiting time (min) | Arrival time (min) |
|---|---|---|---|---|
| $v_1$ | R0 | 4.16 | 0.0 | $6.1^{x}$ |
| | R1 | 4.16 | 10.8 | 16.9 |
| $v_2$ | R2 | 4.23 | 0.0 | $6.2^{x}$ |
| | R3 | 5.78 | 0.0 | 8.5 |

Notes:
[1] The vehicles considered in this scenario departure at time t=0 min
[2] R0, R2: The shortest route calculated by the standard A* algorithm
[3] R1, R3: The route calculated by the algorithm, MOAAstar–II/Waiting, given a speed of 40 km/h (the distance of R1 equals the distance of R0)
[4] x: It is estimated based on the shortest route without considering the moving obstacles

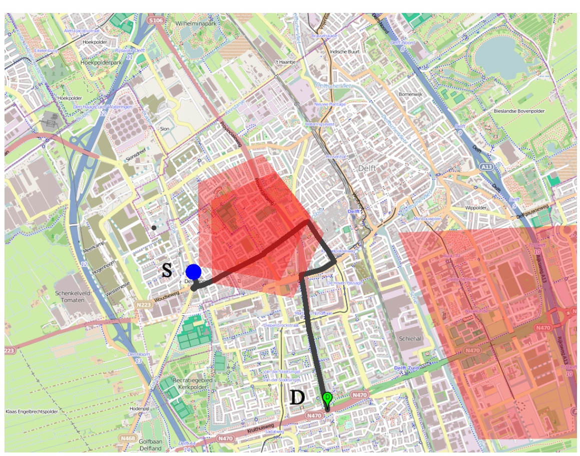
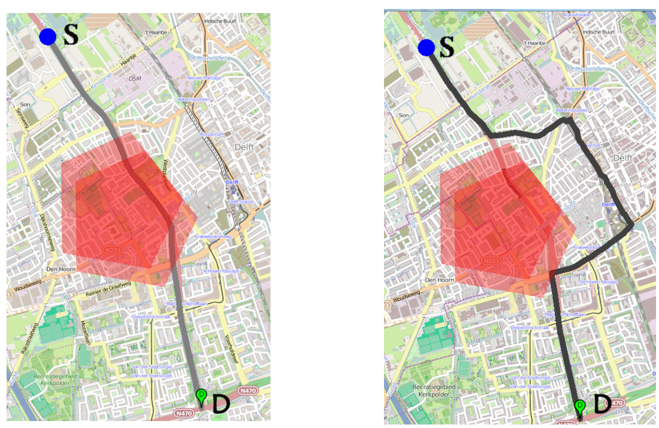



Figure 7.5: Snapshot of calculated routes of vehicle $v_1$ from source $S$ to destination $D$ (t=0 min, $v_1$ is at the source point, the route calculated by algorithm MOAAstar–II/Waiting, R1, has the same geometry as the shortest route, R0)

On the other hand, there is another vehicle $v_2$ that is available and moves at a speed of 40 km/h. Using the proposed system, vehicle $v_2$ gets a route which is different from the shortest route to avoid the obstacles, as depicted in Figure 7.6. As we can see from the table, by following the calculated route, vehicle $v_2$ can reach the destination before the expected arrival time without any waiting, although it is a bit further from the destination, compared to vehicle $v_1$. Because the moving obstacle could cause possible delays during the response, these delays should also be considered during crisis decision making. Taking into account the moving obstacles, the system can not only provide safe and fast routes, but also supports emergency managers in selection of response teams to handle the incidents within the required time limit.




(a) R2, the shortest route

(b) R3, the route calculated by algorithm MOAAstar–II

Figure 7.6: Snapshot of calculated routes of vehicle $v_2$ from source $S$ to destination $D$ (t=0 min, $v_1$ is at the source point)

### 7.3.2 *Case 2: $< M, O, S, M >$navigation of multiple responders to one destination*

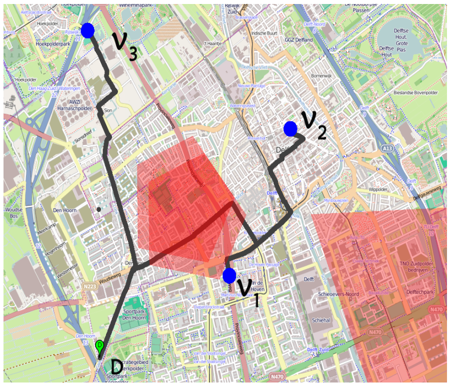
Table 7.4 shows results of application of the system to the case that three responders have to go to the same destination. The involved responders reports their profiles to the system, such as current position, the speed based on the type of vehicles, and departure time. Taking into account the profile of the responders, the system calculates the obstacle avoiding path for each vehicle respectively. As we can see from the table that, only vehicle $v_3$ can arrive at the destination on time, $v_1$ has to spend more traveling time to avoid obstacles, which causes it to arrive about 11.5 min later than the arrival time estimated by the shortest route, and vehicle $v_2$ also has a time delay about 8.8 min. Figure

7.7 shows the snapshots of movements of the involved vehicles towards the same destination. The results reveal that incorporation of predicted data of hazards is essential for estimation of arrival time of relief vehicles, and thus contribute to generation of better emergency plans.

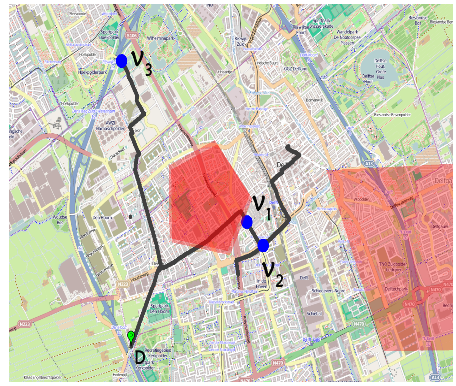
Table 7.4: Calculated results of the case $< M, O, S, M >$

| Vehicle ID | Route ID | Departure time (min) | Distance (km) | Total waiting time (min) | Arrival time (min) |
|---|---|---|---|---|---|
| $v_1$ | R0 | 0.0 | 3.7 | 0.0 | 7.3[x] |
| | R1 | 0.0 | 3.7 | 11.3 | 18.8 |
| $v_2$ | R2 | 1.0 | 5.1 | 0.0 | 8.7[x] |
| | R3 | 1.0 | 5.1 | 8.7 | 17.5 |
| $v_3$ | R4 | 4.0 | 4.1 | 0.0 | 8.8[x] |
| | R5 | 4.0 | 4.1 | 0.0 | 8.8 |

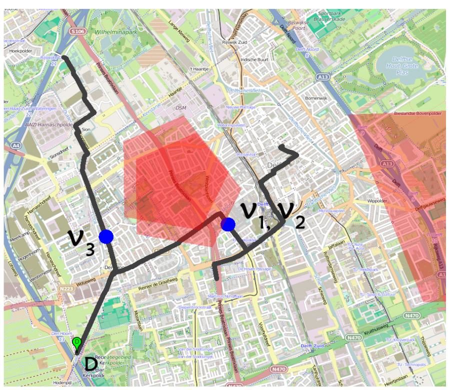
Notes:
1 R0, R2, R4: The shortest routes from different sources to the same destination, calculated by the standard A* algorithm
2 R1: The route calculated by the algorithm, MOAAstar–II/Waiting, given a speed of 30 km/h
3 R3: The route calculated by the algorithm, MOAAstar–II/Waiting, given a speed of 40 km/h
4 R5: The route calculated by the algorithm, MOAAstar–II/Waiting, given a speed of 50 km/h
5 x: It is estimated based on the shortest route without considering the moving obstacles



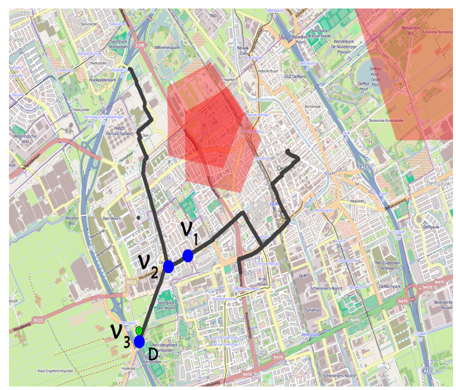

(a) t=0 min, $v_1$ , $v_2$ $v_3$ are at the different source points




(b) t=4 min, $v_1$ is waiting to avoiding moving obstacles, $v_2$ is moving along the calculated route, and $v_3$ is still at the source point




(c) t=7 min, $v_1$ and $v_2$ are waiting at the same location to avoid moving obstacles, and $v_3$ is moving towards the destination




(d) t=16 min, $v_1$ and $v_2$ continue moving towards the destination, and $v_3$ reaches the destination

Figure 7.7: Snapshots of movements of both obstacles (in polygons) and the vehicles (in circle) at different time instances

### 7.3.3 *Case 3: $< O, M, S, M >$navigation of one responder to multiple destinations*

Figures 7.8 depicts a scenario in which five tasks have to be allocated to vehicle $v_1$. Figure 7.9 depicts another scenario that vehicle $v_2$ has to go to also five tasks. The tasks of the same ID in the two scenarios have the same locations, but differ in operation time needed for responders to perform. The operation time of the tasks for each vehicle is listed in Table 7.5. We assume that the two vehicles departure from the same location and move at a constant speed of 40 km/h, and no tasks have a limitation on arrival time. The proposed navigation system plans a trip connecting the locations associated with the involved tasks, minimizing the traveling time based on the operation time of the tasks and the predicted information about the environment affected by the plumes. We compare the route proposed by our algorithm with the optimal route. Table 7.6 and Table 7.7 show the calculated results for the two vehicles respectively. As we can see from the results, vehicle $v_1$ and vehicle $v_2$ get different routes, which both have relative error smaller than 12% and are customized based on their operation time for tasks. Besides, the results also imply the importance of including of the operation time in determining the sequence of tasks that should be followed by the responder, and in the generation of routes that are to avoid moving obstacles.

Table 7.5: The operation time of tasks (min)

| Vehicle ID | $t_o(T_1)$ | $t_o(T_2)$ | $t_o(T_3)$ | $t_o(T_4)$ | $t_o(T_5)$ |
|---|---|---|---|---|---|
| $v_1$ | 2 | 5 | 1 | 4 | 6 |
| $v_2$ | 1 | 4 | 3 | 5 | 1 |

Notes: $t_o(T_j)$ is the operation time of task $T_j$.

Table 7.6: The calculated routes for vehicle $v_1$

| | The direction of the route | Total travel time (mins) |
|---|---|---|
| Proposed route | $S \to T_1 \to T_5 \to T_4 \to T_3 \to T_2$ | 35.6 |
| Optimal route | $S \to T_3 \to T_4 \to T_5 \to T_1 \to T_2$ | 32.0 |
| Relative error | 11% | |

Table 7.7: The calculated routes for vehicle $v_2$

| | The direction of the route | Total travel time (mins) |
|---|---|---|
| Proposed route | $S \to T_1 \to T_2 \to T_5 \to T_3 \to T_4$ | 31.9 |
| Optimal route | $S \to T_3 \to T_4 \to T_5 \to T_1 \to T_2$ | 29.0 |
| Relative error | 10% | |

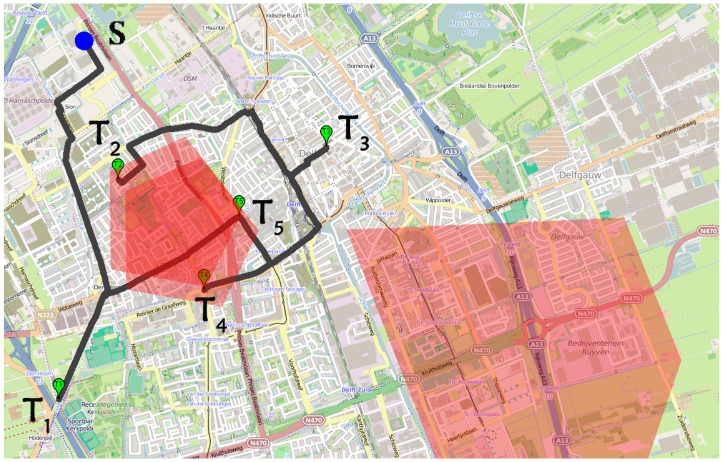



Figure 7.8: Snapshot of calculated routes for vehicle $v_1$ ($t = 0$ min, the vehicle is at the source point)

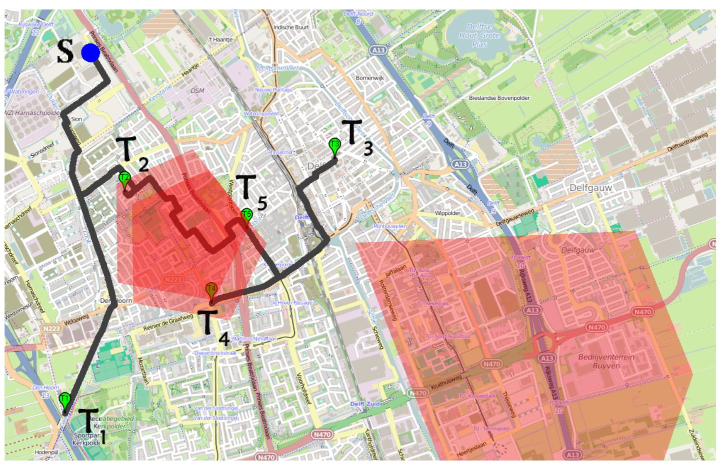



Figure 7.9: Snapshot of calculated routes for vehicle $v_2$ ($t = 0$ min, the vehicle is at the source point)

### 7.3.4 *Case 4:* $< M, M, S, M >$*navigation of multiple responders to multiple destinations*

In this subsection, we study the navigation case that include multiple responders and multiple destinations. In this case, each destination only needs to be visited by one responder. Figure 7.10 and 7.11 depicts the same situation that involves 3 relief vehicles and 7 assistance tasks, $|V| = 3, k = 7$, but differ on the penalty weights that are applied to the calculation. We assume the involved vehicles move at the same speed of 40 km/h, and the operation time of all tasks is 1 min. The system calculates routes for the involved vehicles using Sequential Single-Item (SSI) Auctions presented in Section 6.6. As previously mentioned, minimizing the number of delays is considered as our primary objective and followed by minimizing the total amount of time delays. Thus, for the multi-objective optimization of path $P_i$ of vehicle $v_i$, the largest penalty weight should be imposed on $N(v_i, P_i)$ in order to direct the search towards solutions with less number of delays. Based on the results of experiments with random destinations, in this scenario we used the following weights: $w_1 = 10000$, $w_2 = 1000$. Table 7.8 shows the calculated results. The generated results are as follows: 1) path $P_1$ of vehicle $v_1$: $S_1 \rightarrow T_3 \rightarrow T_6 \rightarrow T_2$; 2) path $P_2$ of vehicle $v_2$: $S_2 \rightarrow T_4 \rightarrow T_5$; 3) path $P_3$ of vehicle $v_3$: $S_3 \rightarrow T_7 \rightarrow T_1$. We also apply our algorithm to this case without using penalty weights ($w_1 = 0$, $w_2 = 0$). The generated results are as follows (see Table 7.9): 1) path $P_1$ of vehicle $v_1$: $S_1 \rightarrow T_3 \rightarrow T_6$; 2) path $P_2$ of vehicle $v_2$: $S_2 \rightarrow T_4 \rightarrow T_5$; 3) path $P_3$ of vehicle $v_3$: $S_3 \rightarrow T_7 \rightarrow T_1 \rightarrow T_2$. From these tables, we can see that the max delay, $\max_{i=1}^{|V|} DT(v_i, P_i) = 6.3$ min, in Table 7.8 is slightly higher than $\max_{i=1}^{|V|} DT(v_i, P_i) = 4.9$ min in Table 7.9. However, by applying the penalty weights, the max number of delays $\max_{i=1}^{|V|} DN(v_i, P_i)$ is reduced from 2 to 1.

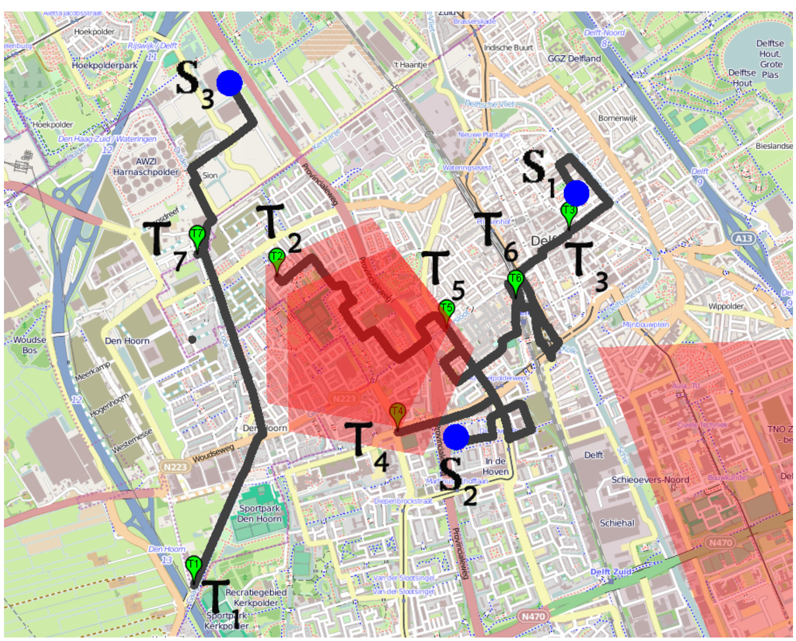



Figure 7.10: Snapshot of calculated routes for 3 vehicles with 7 tasks ($w_1 = 100000$, $w_2 = 1000$, t=0 min, all vehicles are at the source points)

Table 7.8: Calculated results using the penalty weights ($w_1 = 100000$, $w_2 = 1000$)

| Vehicle ID | $v_1$ | | | $v_2$ | | $v_3$ | |
|---|---|---|---|---|---|---|---|
| Task | $T_3$ | $T_6$ | $T_2$ | $T_4$ | $T_5$ | $T_7$ | $T_1$ |
| $t_A$ (min) | 5.6 | 9.4 | 17.3 | 6.2 | 13.6 | 5.2 | 9.4 |
| $t_{\text{RA}}$ (min) | 20 | 16 | 11 | 8 | 9 | 12 | 8 |
| $t_{\text{delay}}$ (min) | 0 | 0 | 6.3 | 0 | 4.6 | 0 | 1.4 |

Notes:
[1] $t_{\text{A}}$ is the actual arrival time of the task
[2] $t_{\text{RA}}$ is the required arrival time of the task
[3] $t_{\text{delay}}$ is the time delay between $t_{\text{A}}$ and $t_{\text{RA}}$ of the task

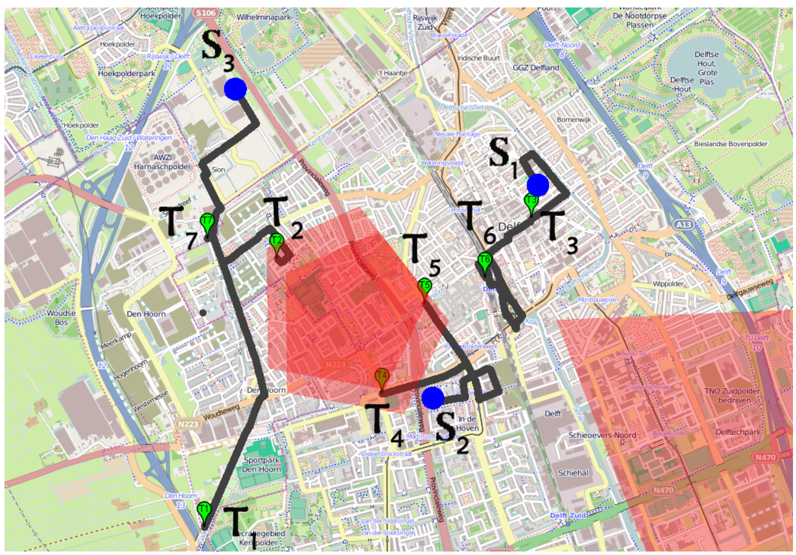



Figure 7.11: Snapshot of calculated routes for 3 vehicles with 7 tasks ($w_1 = 0$, $w_2 = 0$, t=0 min, all vehicles are at the source points)

Table 7.9: Calculated results without using the penalty weights ($w_1 = 0$, $w_2 = 0$)

| Vehicle ID | $v_1$ | | | $v_2$ | | $v_3$ | |
|---|---|---|---|---|---|---|---|
| Task | $T_3$ | $T_6$ | $T_4$ | $T_5$ | $T_7$ | $T_1$ | $T_2$ |
| $t_A$ (min) | 5.6 | 9.4 | 6.2 | 13.6 | 5.2 | 9.4 | 14.5 |
| $t_{\text{RA}}$ (min) | 20 | 16 | 8 | 9 | 12 | 8 | 11 |
| $t_{\text{delay}}$ (min) | 0 | 0 | 0 | 4.6 | 0 | 1.4 | 3.5 |

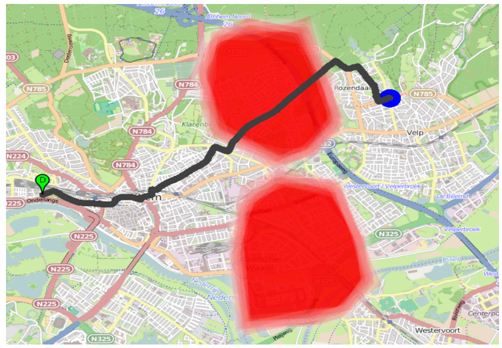
Notes:
[1] $t_{\text{A}}$ is the actual arrival time of the task
[2] $t_{\text{RA}}$ is the required arrival time of the task
[3] $t_{\text{delay}}$ is the time delay between $t_{\text{A}}$ and $t_{\text{RA}}$ of the task

## 7.4 Application to the case with uncertain obstacles

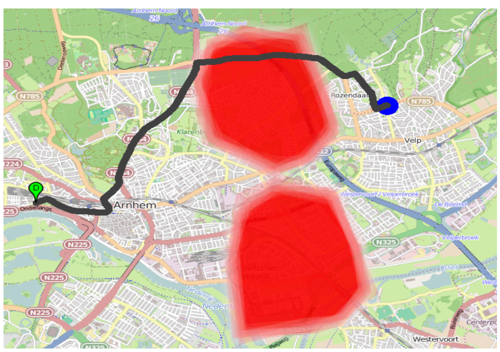
In this section, we test the system in navigation in the presence of uncertain moving obstacles, using the road network dataset in Arnhem, The Netherlands. The road network is comprised of 13336 road segments and 11712 road junctions. We suppose that two moving plumes are moving across the city. A group of datasets of polygons with timestamps are generated following a given normal distribution of positions to simulate the movement of plumes. Our navigation system fetches the data of obstacles, and performs spatial analysis to compute the likelihood of road segments and junctions being affected by the obstacles. Here we assume that the obstacles have the same concentration, and define 6 levels of risk as follows: $L0$= $[0, 0.02]$, $L1 = [0.02, 0.05]$, $L2$= $[0.05, 0.1]$, $L3$= $[0.1, 0.3]$, $L4 = [0.3, 0.6]$, $L5$= $[0.6, 1]$. The information of risk levels of road segments and junctions is stored in the database according to the data model CDM–C/Uncertainty (presented in Section 4.4.3). We consider the following two scenarios: 1). The responders have the same protective equipment but different amounts of oxygen (see Section 7.4.1); 2). The responders have the same amount of oxygen but different protective equipment (see Section 7.4.2). The algorithm, MOAAstar–III/Uncertainty (presented in Section 6.4), is used to generate the customized routes, based on the profile of responders. The calculated results are displayed on the 2D map.

### *7.4.1 Scenario 1: navigation for responders with different amounts of oxygen*

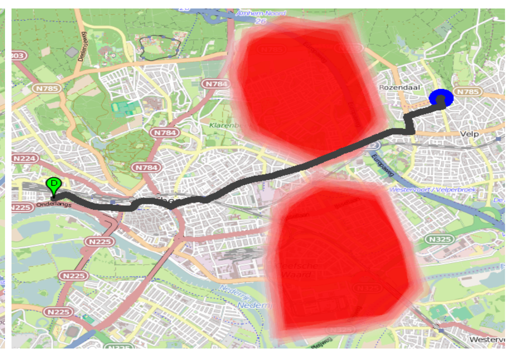
In this scenario, the system calculates the safe routes for 3 vehicles that have to go from the same source and destination points, given the same speed 30 km/h and the same categorization of risk levels. We suppose that the responders in different vehicles have different amounts of oxygen, which is indicated by $r_{max}$, i.e., the maximal amount of time for passing through the obstacles. Figure 7.12 shows a comparison of the results calculated by our system considering different $r_{max}$. As we can see from the figure, although the emergency task in this scenario requires all responders to go to the same destination, the system, which performs the calculation with the given $r_{max}$ of each vehicle, generates different routes for them. The calculated results are shown in Table 7.10. As shown in the table, our developed algorithm is capable of generating routes that have the total risk constrained by the user-specified $r_{max}$. With a higher $r_{max}$, the responders are allowed to stay longer in the obstacles, and thus follow a shorter route to reach the destination. The scenario also shows that, following different routes, the responders would have different amounts of risk accumulated along their routes. If $r_{max}$ is not considered in the routing, the total risk of the route that the responders could confront could be larger than their acceptable value, which would slow or even endanger the their response process.



(a) Vehicle $v_1$, $r_{max} = 0$ min



(b) Vehicle $v_2$, $r_{max} = 2$ min



(c) Vehicle $v_3$, $r_{max} = 7$ min

Figure 7.12: Snapshots ($t = 0$ min) of routes calculated for 3 vehicles (in circle) with different $r_{max}$. The vehicles are at the source points. The shapes and positions of the obstacles (in polygons) will change as the vehicles are moving towards the destinations.

Table 7.10: Calculated results considering $r_{max}$

| Vehicle ID | $r_{max}$ (mins) | Total Risk (mins) | Distance (km) | Total waiting time (mins) | Arrival time (min) |
|---|---|---|---|---|---|
| $v_1$ | 0 | 0.0 | 7.5 | 2.6 | 18.2 |
| $v_2$ | 2 | 0.5 | 8.2 | 0.0 | 16.7 |
| $v_3$ | 7 | 6.0 | 6.6 | 0.0 | 14.8 |

Notes:
[1] Vehicles $v_1$, $v_2$ and $v_3$ have the same source and destination
[2] The routes are calculated by the algorithm, MOAAstar–III/Uncertainty, given a speed of 30 km/h and a departure time t = 0.0 min
[3] The routes are calculated based on the same categorization of risk levels: *full-safe levels*={L0, L1, L2}, *partial-safe levels*={L3, L4, L5}, *non-safe levels* = ∅

### 7.4.2 *Scenario 2: navigation for responders with different protective equipment*

As responders may have different protective equipment, different classifications of risk levels would be made by them and used in the routing process. In this scenario, we compare the relief routes calculated based on the different categorization of risk levels. The considered responders move at the same speed, 50 km/h, and have the same $r_{max}$ = 6 min. Figure 7.13 depicts 3 routes calculated for 3 vehicles. These routes have the same pair of source and destination, but are generated based on different categorization of risk levels. Table 7.11 shows the results of the calculated routes. As shown in the table, the total risk of routes are below the given constrain. The route generated for vehicle $v_1$ has the highest risk value, but it is still acceptable for $v_1$ and allows it to reach the destination in a least amount of travel time among the three vehicles. This is because that $v_1$ accepts higher risk-levels (as shown in Figure 7.13), which makes it possible to pass through the obstacles that are non-safe for the other vehicles. Waiting option is used by vehicle $v_2$ to avoid the non-safe obstacles and to reduce the risks. With our navigation system, responders can get customized routes, using their own classification of risk levels based on their available protective equipment.

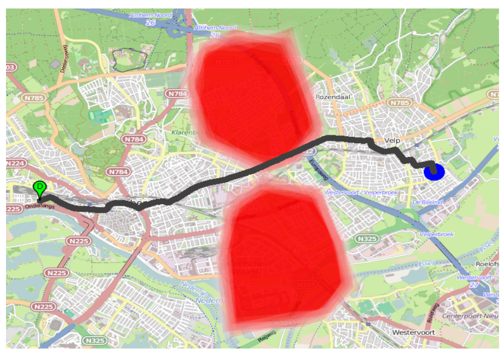
Table 7.11: Calculated results considering different categorization of risk levels

| Vehicle ID | $r_{max}$ (mins) | Total Risk (mins) | Distance (km) | Total waiting time (mins) | Arrival time (min) |
|---|---|---|---|---|---|
| $v_1$ | 6.0 | 5.9 | 7.6 | 0.0 | 17.1 |
| $v_2$ | 6.0 | 0.4 | 8.7 | 0.6 | 19.3 |
| $v_3$ | 6.0 | 0.0 | 9.5 | 0.0 | 19.4 |

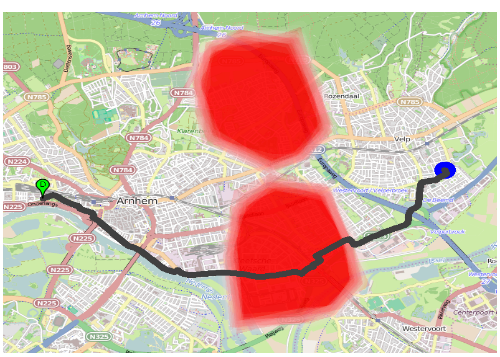
Notes:

[1] Vehicles $v_1$, $v_2$ and $v_3$ have the same source and destination, and move at the same speed of 30 km/h

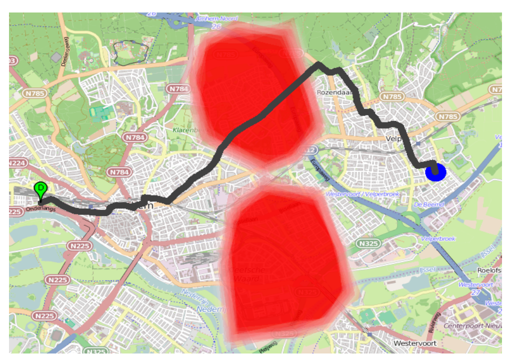
[2] The routes are calculated by the algorithm, MOAAstar–III/Uncertainty, given $r_{max}$ = 6 min and a departure time t = 0.0 min



(a) Vehicle $v_1$, *full-safe levels*={L0, L1, L2}, *partial-safe levels*={L3, L4, L5}, *non-safe levels* = ∅



(b) Vehicle $v_2$, *full-safe levels*={L0, L1}, *partial-safe levels*={L2, L3}, *non-safe levels* ={L4, L5}



(c) Vehicle $v_3$, *full-safe levels*={L0}, *partial-safe levels*={L1}, *non-safe levels* ={L2, L3, L4, L5}

Figure 7.13: Snapshots ($t = 0$ min) of routes calculated for 3 vehicles (in circle) with different categorization of risk levels. The vehicles are at the source points. The shapes and positions of the obstacles (in polygons) will change as the vehicles are moving towards the destinations.

## 7.5 Concluding remarks

In this chapter, we have applied the prototype system to a variety of cases. The experimental results indicate that our data models can manage various types of spatio-temporal data, reflect the dynamics of the road network during disasters, and allow relevant data to be appropriately organized to facilitate automated network analysis and dynamic simulation. The application results also show that the extended algorithms, incorporating the dynamic data of hazards and the information of users, provide safer routes to the destinations, highlighting the importance of the hazard model and the user profile in emergency planning. As demonstrated by the experiments we carried out, our navigation system can not only provide safe routes for one or multiple responders towards one or many destinations, but also generate personalized routes according to the profile of responders, offering a promising direction for a wider range of applications.

# 8 CONCLUSIONS AND FUTURE WORK

This chapter summarizes the main achievements and the limitations of our work, and presents an outlook on future research. In Section 8.1, we provide our answers to the research questions posed in this PhD study. Section 8.2 gives some discussions on the proposed navigation system. Finally, we conclude with a list of the suggested topics that can be investigated in future research (see Section 8.3).

## 8.1 Conclusions

With the frequent natural or man-made disasters occurring in recent years, emergency navigation for first responders poses a set of serious challenges for researchers in the navigation field. For one thing, emergencies caused by disasters can result in both static and moving obstacles in the infrastructure. For another, during the disaster response, the rescuers need to go to a number of locations to carry out their relief tasks. Therefore it is crucial to investigate the navigation for first responders among moving obstacles. By using hazard simulations, the agent technology, and geo-DBMS, this thesis has contributed to developments of solutions for path planning for first responders in the presence of moving obstacles.

With the objective in this PhD research, we defined the main question as follows:

*How do we navigate one or more first responders to one or multiple destinations avoiding moving obstacles?*

Within the above main research question, the following 4 sub-questions have been formulated in Section 1.2. We summarize our answers to those questions as follows:

1. *What navigation cases need to be considered for assisting first responders?*
   In Chapter 3, we have provided an extended taxonomy of navigation

for first responders, which categorizes navigation cases on basis of type and multiplicity of first responders, destinations, and type of obstacles. Within this classification, we have identified 16 navigation cases. Our investigation on these cases reveals that the navigation cases involving the moving obstacles in the road network still need to be addressed. Among these cases, four navigation cases, i.e., 1) $< O, O, S, M >$; 2) $< M, O, S, M >$; 3) $< O, M, S, M >$; 4) $< M, M, S, M >$, have been selected as the starting points and studied in this thesis. The findings from the studies of these four cases will provide us with insights into the investigation of the left cases.

2. *What data models should be developed to support path planning among moving obstacles?*
   Based on the requirements presented in Section 4.3, we have defined three versions (A, B and C) of spatial data models, extending the model proposed by Dilo and Zlatanova (2011) to support path planning and evaluation. By taking some classes from the model in Dilo and Zlatanova (2011), data model, CDM–A/Predictions, was introduced in Section 4.4.1 and designed for the management of disaster-related information essential for routing. This data model captures both static information, such as the type of the incident, the topology of the road network, and dynamic information, such as the changing state of road junctions, and the position of the moving obstacles. Both the predicted data (e.g., the closed time of roads) and real-time data (i.e, the speed of vehicles) have been structured in the model to support calculation of the obstacle avoiding routes. Data model, CDM–B/Multi-destination, is an extension of model CDM–A, and covers information related to the relief routes for responders with multiple emergency tasks (see Section 4.4.2). For dealing with the uncertainty of the obstacles generated from hazard simulations, data model, CDM–C/Uncertainty, which was derived based on CDM–A and CDM–B, supports storage of the uncertain obstacles and representation of the risk of components of the road network. In CDM–C, we introduced the concept, *risk level*, to describe the status of road segments and junctions affected by the uncertain moving obstacles (see Section 4.4.3). The experimental results indicate that our data models can manage various types of spatio-temporal data, reflect the dynamics of the road network during disasters, and allow relevant data to be appropriately organized to facilitate automated spatial analysis and dynamic simulation.

3. *What types of agents are needed to assist path planning among moving obstacles?*

In Chapter 5, we have presented a multi-agent based navigation system, aiming to assist path planning in the presence of moving obstacles caused by different types of hazards. In connection with geo-DBMS, a set of software agents is designed and developed within the system: 1). Hazard Agent which supports collection and transformation of information from hazard simulations; 2). Task Monitoring Agent which monitors the state of emergency tasks; 3). Vehicle Monitoring Agent which tracks the movement of vehicles; 4). Network Monitoring Agent which collects spatio-temporal data of the road network; 5). Task Allocation Agent and Vehicle Agent which perform route calculation in dynamic environments affected by disasters. While focusing on their own tasks, these different types of agents communicate with each other and coordinate their tasks using the communication facilities provided by the agent technology, which enables the automatic planning or re-planning of routes. More importantly, by distributing the route computation to a group of agents, the proposed multi-agent system is able to handle the navigation problems that involve multiple responders and multiple destinations, which are NP-hard problems. The use of the agent technology not only supports generation of safe routes, but also enables the testing and evaluation of the developed data models and routing algorithms. Using the agent simulation tool, we also developed special agents to represent the responders to simulate their movement, which allows us to assess the performance of the designed models and algorithms.

4. *What algorithms should be developed for path planning among moving obstacles?*
   Three types of path planning problems are distinguished in this part of the research: one-to-one, one-to-many, many-to-many. For addressing these problems, in Chapter 6 new algorithms have been proposed as well as integrated with other existing algorithms. Regarding the one-to-one path planning, three versions (I, II and III) of the modified A* algorithms have been developed to calculate obstacle-avoiding routes, considering the speed of vehicles, departure time, and the predicted information about the state of the road, etc. On the basis of the A* search mechanism, the first algorithm, MOAAstar–I/Non-waiting, uses the predicted availabilities of roads to generate routes avoiding moving obstacles. It can be applied to situations (e.g., fires) in which the nodes and edges will not be available once they are affected by the obstacles (see Section 6.2). Extending MOAAstar–I, the second algorithm, MOAAstar–II/Waiting, was developed, using the state defined based on the open intervals of the road network. It introduces waiting options to avoid moving obstacles in

certain disastrous events, like plumes, taking into account the availabilities of both road segments and junctions (see Section 6.3). Considering that responders may have some protective equipment that allows them to pass through certain types of obstacles (e.g., plumes), in Section 6.4 we presented our third algorithm, MOAAstar–III/Uncertainty, which can incorporate the user profile into the routing in the presence of the uncertain moving obstacles. For navigating one responder to multiple destinations, we used insertion heuristics for optimizing the route visiting the given set of destinations, and designed a multi-objective cost function with the aim to make a balance between the number of delays, the sum of all delays, and the task-completion time. To solve many-to-many path planning problem, we employed sequential single-item (SSI) auctions, to assign the target locations to the rescue vehicles, using the two types of agents (i.e., Task Allocation Agent and Vehicle Agent). We have applied the developed algorithms to a number of navigation scenarios. The results of the application demonstrate their ability of generating safe routes for one or multiple responders towards one or many destinations.

## 8.2 Discussions

Aiming at providing safe and feasible routes for responders in the presence of moving obstacles, our multi-agent based navigation system has the following features:

- Support integration of hazard simulation models for emergency navigation. The hazard models are used to provide the predicted information about moving obstacles;
- Support management of various types of spatio-temporal data of road networks affected by hazards;
- Support generation of safe routes for individual responders as well as multiple responders, taking into account the dynamic data of moving obstacles;
- Support incorporation of the profile of responders into the routing process. The responders can classify the risk levels to define the state of road networks, according to their available protective equipment;
- Support the path planning among the uncertain moving obstacles. The Monte Carlo simulation method is applied to quantify the influence of the uncertain obstacles on the road network.

Although the system has shown its capabilities to support navigation among moving obstacles, there are still some limitations that may affect the use of the proposed system in real disaster situations and need to be considered in further developments.

First of all, there is not yet a direct connection between our system and the hazard simulation model. Because we need only the output data from the hazard simulation, currently we assume that these data have been provided by external software or a simulation system and stored in the database. The integration of the hazard model into the system on board of vehicles could facilitate the computation and support real-time navigation. This is quite important when there is a limited or no bandwidth connection available during the disaster response. Using the hazard models in the navigation system, the responders on the field can send the real-time data of the environment (e.g., the wind direction, wind speed) directly to their mobile devices to generate the new predictions of hazards for routing, instead of spending long time in transferring the data back to the emergency response center.

Second, in the proposed system architecture, currently we only integrate the simulation model of hazards to provide the predictions of obstacles, and did not consider the movement of crowds. In the case of large-scale disasters, the blockage of roads and streets due to the spread of hazards can also influence the route selection of pedestrians and car drivers, and cause traffic congestion in transportation networks, which would slow down the disaster response. Therefore, it is necessary to make the proposed system linked with models that are capable of simulating the human behaviors and estimating the future traffic flows during disasters. With these models, we can derive the speed at which the rescue vehicles can be travelling, which contributes to the generation of effective route plans. Besides, for addressing the dynamic targets, models that can provide the predicted movement of the pursued objects also need to be incorporated into the system to help identify the possible interception points.

Third, currently the routes are calculated in an automatic way and we assume that the drivers follow the calculated routes. The present developed system does not consider the involvement of vehicle drivers. However, in some situations, the routes generated by the system could not meet the responders' requirements, and the responders may want to interact with the system and influence calculation results. For example, in the case of smoke plumes, the plumes could affect the rescue point shortly after the arrival of responders, which result in no enough time for them to perform their tasks. In this case, the drivers need to make their own decisions on route choices, and the system should be able to take driver behaviors into account and to adjust the planned routes according to actual situations. To support this interaction between drivers and the navigation system, adjustable autonomy techniques would be needed.

Fourth, our current data models only handle data that are essential for emergency navigation. The structuring of the OSM data and the hazard simulation output data used by our system is not considered in the models. It should also be noted that, the developed models do not distinguish situations that whether the road is partially affected or fully affected. Because the responders can also wait at a certain point in the roads to avoid obstacles instead of in the junctions, special approaches would be needed to generate paths in accordance with the actual movement of vehicles, for example, introducing more points along the road to create a more detailed model of road networks; using the length of affected road parts combined with the timestamp for route calculation. Another major concern is the accuracy of the input data, such as the data of the road network, estimated operation time of emergency tasks, and real-time position of relief vehicles, etc. Uncertainties and errors that arise from collection and generation of these data play a special role in the route determination. Therefore, extensions of data models for description of the uncertainties in these data would be necessary to support generation of safe routes for responders.

Last but not the least, in this thesis we have developed three versions of the modified A* algorithms to address the navigation cases with moving obstacles, particularly the cases of plumes and fires. However, in a sense there is no general routing algorithm that can cover all possible navigation situations, and more algorithms would be needed and developed for navigation for responders. This is because that many situations could occur during the response to disasters, which leads to a variety of navigation services requested by responders. For example, depending on the available protective equipment, responders may have different objectives for optimization of the path planning problems, such as maximizing the safety, minimizing the total risk, minimizing the total travel time; as certain tasks require cooperation between response groups moving from different locations, the involved vehicles have to be routed to the same destination at the same time to perform these tasks. To be able to deal with these situations, a set of new routing algorithms should be developed, which enables the responders to select some of the algorithms that would fulfill their requirements in real situations.

## 8.3 Recommendations for further research

For future research, many issues related to emergency navigation need to be investigated. Along with these investigations, several extensions should be studied to enhance the routing capability of the current approach.

### MORE NAVIGATION CASES

As presented in Wang and Zlatanova (2013c), there are still a couple of navigation cases that need to be investigated, especially the ones that involve dynamic destinations. For example, one moving object has to be routed to many dynamic destinations, avoiding many moving obstacles. Another research direction would be to explore further some extreme cases, for instance, the target point could be affected by the obstacles during the course of an incident, resulting in no available route until the incident is over. Currently we assume that the responders have enough amount of time for them to safely carry out their tasks at destinations, but it could also happen that there is no enough time for them to finish the tasks. In this case, the responders have to decide either to wait outside the dangerous areas or to work under hazardous conditions. To support that, the navigation system should be able to generate appropriate routes based on the decisions made by responders.

### APPLICATION TO VARIOUS TYPES OF HAZARDS

As shown in Chapter 7, we applied our system to aid navigation in the case of forest fires and smoke plumes. But our approach is not limited to route planning during these hazardous events, but can be extended to assist navigation among moving obstacles brought about by other types of disasters (e.g., floods). For supporting navigation in these different types of natural disasters, hazard simulation models that can produce reliable prediction of the spread of hazards are required. Because these simulation models differ on the output format and use different local coordinate systems, more types of agent would be needed and integrated into the system to handle heterogeneous data from these models and to extract information essential for navigation purpose. Moreover, some extensions of our data models would also be needed to merge and organize information from the hazard models and to meet a wider range of informational needs when multiple disasters occur simultaneously.

### MAP VISUALIZATION AT DIFFERENT LEVELS OF DETAILS

Maps play a special role in a disaster response, supporting decision making in different levels (from strategic to operational). During navigation for responders among moving obstacles, different maps would be used, which requires visualization of spatial data at different levels of detail. For example, for responders on the field, a map with detailed representation of environments is required, guiding them to avoid dangerous areas; for emergency managers in the CCC, an aggregated map that provides an overview of disaster situations would be needed, supporting route planning at the strategic level. Both 2D and 3D visualization techniques will be used to support displaying of the calculated routes, obstacles, and the affected environments. As large volume of data of hazards could be received from Command and Control Center, variable-scale methods (Meijers, 2011) can be used to facilitate the retrieving and visualization of the data. This would allow responders to easily view the objects that are relevant to their tasks by zooming in or zooming out to the desired scale.

### ANALYSIS OF THE ALGORITHMS

The analysis of the algorithms is important when we move the route calculation to mobile devices. On one hand, the mobile device has limited computational resources that can support running of the algorithms, which could result in more computation time to generate results. On the other hand, during a disaster response there is no time to wait for hours calculations. By analyzing the algorithms, the developer of emergency navigation systems can estimate how much memory and computation overhead would be required, which can help them to choose the hardware that is capable of running the algorithms and quickly generating results. Furthermore, sensitive analysis of the algorithms will also be needed. In Chapter 6, a set of parameters has been introduced and used to configure the algorithms. For example, in MOAAstar–III/Uncertainty, we use time step $\Delta t$ to discretize the waiting time series; in the one-to-many algorithm, the penalty weights are applied to deal with different objectives. By conducting sensitive analysis, developers can establish standards on the configuration of these parameters in the algorithms, which make the algorithms able to generate routes according to real situations and the profile of responders.

### DIFFERENT OPTIMIZATION TECHNIQUES FOR ROUTING

Because many navigation problems involved in the disaster response are NP-hard problems, it is also necessary to investigate the use of different optimization techniques, such as genetic algorithms, ant colony algorithms, and particle swarm algorithms. By using heuristics, these algorithms can generate the results that are close to the optimal solutions in a short time. Comparisons between these algorithms in different aspects (e.g, computation time, the optimality) would also be made to guide the selection of these algorithms for different navigation purposes.

### EXTENSION OF THE TAXONOMY

As we mentioned in Chapter 3, the taxonomy we proposed in this thesis is not yet complete. Further refinements can be made by adding more criteria, e.g. the movement of the obstacle can be a priori known or not; the obstacle can have either distinct boundaries or fuzzy shape due to the nature of disasters. Such a taxonomy can not only help people to understand the differences and characteristics of navigation cases during disasters, but also facilitate the standardization of all relevant aspects of navigation for first responders, which contributes to the development of emergency navigation systems.

#### TRACKING AND MONITORING

One of the integral functions that the navigation system should provide is to track and monitor both the vehicles and the moving obstacles. Real-time information about these two types of moving objects are of importance in improving the planned routes. For one thing, when a rescue vehicle follows a specific route towards its destination, the environment can change rapidly and the drivers may encounter moving obstacles that are not known beforehand. A live connection between sensors and the simulation models should be built to enable the responders to send real-time measurements to the models in the control center or system on board to generate new predictions of the obstacles. To support collection of real-time measurements, the data model needs to be adjusted to structure the information from the sensors. Besides, the speed of the vehicles is also an important aspect, which can be influenced by many factors, such as the weather conditions, slow traffic, congestions and road blockages. By tracking the position of the responders, the system can derive the real-time speed of the vehicles and introduce it into the routing process, correcting navigation with respect to the real conditions. In relation to this, a rule-based approach should be defined and applied to analyze whether and when new routes or new speed instructions are needed.

#### USING PARALLEL COMPUTING FOR ROUTING

The parallel computing technique can be useful for emergency navigation in various aspects. For example, it can support parallel processing of spatial-temporal data, especially when we deal with large road networks. As different constrains can be applied in generation of routes in the presence of moving obstacles, which could increase the computation complexity, the parallel computing technique can also accelerate the route calculation by using the power of multi-core processor.

#### APPLICATION OF OUR APPROACH TO OTHER AREAS

Although we focus on the navigation for responders, the approach developed in this research can be applied to other areas. For example, using the predicted information structured in our geo-DBMS, the emergency managers can define the evacuation areas that could be affected by hazards, and make evacuation plans for refugees. Various spatial queries can be performed to obtain the information needed for rescue operations (e.g., how many police cars have to be sent to alert the people in the threaten areas, how many houses at this moment are in the obstacles). The recorded routes can also be analysed after disaster response to evaluate routing decisions made by the responders. What's more, the proposed approach can be applied to situations in daily life. For instance, the developed algorithms can be used for route selection in traffic networks, where people should be guided to avoid the traffic congestion. They are also capable of planning routes for tourists who visit different booths in an exhibition hall, meanwhile avoiding the crowds.

### INTEGRATION OF INDOOR AND OUTDOOR NAVIGATION

As many disasters occur in the built environment with all kinds of complex multi-level structures, responders may have to move though both the outdoor and indoor space. Therefore, there is a need for a system that is capable of seamlessly integrating indoor and outdoor navigation. To support this integration, new data models are required. They should be table not only to represent the indoor and outdoor environments, but also to deal with changes that are caused by disasters, such as current availability of exits, the status of stairs. Besides, it is also necessary to investigate the developments of algorithms that can be used for multi-modal (e.g, car, bus, walking) route planning based on the proposed models. As at this point, the research on indoor modeling and LBS is still at the early stage, there are many issues that are open and need to be investigated to make this integrated navigation practical for real life situations.

### DEVELOPMENT OF ANDROID APPLICATION ON MOBILE DEVICE

Using standard Web services, an Android navigation application, which can connect to the Control Center to get navigation assistance, should be developed and tested in both the daily practice and real disasters. Because the connection to the Control Center could be lost during disasters, especially in extreme weather conditions, this application should also be capable of storing the data of moving obstacles, and carrying out the route computation locally on the mobile device. As responders often work in groups and cooperate with each other, the communication functionality should be implemented in the device to allow responders to share and update the information on the field. Another important component in the application is the user interface. Various styling options will be provided in the user interface for different situations, e.g., waiting and moving, day and night, and urgent and non-urgent.

NAVIGATION IN THE ENVIRONMENT CONTAINING BOTH ROAD NETWORKS AND FREE SPACE

Because some incidents could happen in the places where there is no road network around them, there is also a need to investigate the navigation in the environment that involves both road networks and free space. Special routing algorithms should be developed to deal with the obstacles moving in both types of environment. Both the vehicle type and the terrains would be considered to generate routes that fit the profile of responders. The approaches that are proposed in this research topic can be useful for many applications, for example, the integration of outdoor and indoor navigation.

# BIBLIOGRAPHY


Ale, B., van Gulijk, C., Hanea, A., Hanea, D., Hudson, P., Lin, P.-H., and Sillem, S. (2014). Towards {BBN} based risk modelling of process plants. *Safety Science*, 69(0):48 – 56. (Cited on page 3).

Ale, B. J. (2013). Better science does not make decisions easier. *Safety, Reliability and Risk Analysis: Beyond the Horizon*, pages 3–11. (Cited on page 3).

Allen, G. (2007). Building a dynamic data driven application system for hurricane forecasting. In *Computational Science–ICCS 2007*, pages 1034–1041. Springer. (Cited on page 20).

Aydinoglu, A. C., Yomralioglu, T., Quak, W., and Dilo, A. (2009). Modeling emergency management data by UML as an extension of geographic data sharing model: ASAT approach. In *Proceedings of TIEMS 2009 annual conference*. TIEMS society. (Cited on page 17).

Ayre, K. K. and Landis, W. G. (2012). A bayesian approach to landscape ecological risk assessment applied to the upper grande ronde watershed, oregon. *Human and Ecological Risk Assessment: An International Journal*, 18(5):946–970. (Cited on page 23).

Bădică, C., Budimac, Z., Burkhard, H.-D., and Ivanović, M. (2011). Software agents: Languages, tools, platforms. *Computer Science and Information Systems/ComSIS*, 8(2):255–298. (Cited on page 9).

Balasubramanian, V., Massaguer, D., Mehrotra, S., and Venkatasubramanian, N. (2006). Drillsim: A simulation framework for emergency response drills. In Mehrotra, S., Zeng, D., Chen, H., Thuraisingham, B., and Wang, F.-Y., editors, *Intelligence and Security Informatics*, volume 3975 of *Lecture Notes in Computer Science*, pages 237–248. Springer Berlin Heidelberg. (Cited on page 29).

Bein, D., Madan, B. B., Phoha, S., Rajtmajer, S., and Rish, A. (2013). Dynamic data-driven sensor network adaptation for border control. In *Proceedings of*

*SPIE - The International Society for Optical Engineering*, volume 8711, pages 87110J–87110J–9. (Cited on page 20).

Belkhouche, F., Belkhouche, B., and Rastgoufard, P. (2007). Parallel navigation for reaching a moving goal by a mobile robot. *Robotica*, 25(1):63–74. (Cited on page 16).

Bellifemine, F., Bergenti, F., Caire, G., and Poggi, A. (2005). Jade – a Java agent development framework. In *Multi-Agent Programming*, pages 125–147. Springer. (Cited on page 102).

Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90. (Cited on page 32).

Bordini, R. H., Braubach, L., Dastani, M., El Fallah-Seghrouchni, A., Gomez-Sanz, J. J., Leite, J., O'Hare, G. M., Pokahr, A., and Ricci, A. (2006). A survey of programming languages and platforms for multi-agent systems. *Informatica (Slovenia)*, 30(1):33–44. (Cited on page 9).

Breunig, M. and Zlatanova, S. (2011). 3D geo-database research: Retrospective and future directions. *Computers* & *Geosciences*, 37(7):791–803. (Cited on page 24).

Buford, J. F., Jakobson, G., and Lewis, L. (2006). Multi-agent situation management for supporting large-scale disaster relief operations. *International Journal of Intelligent Control and Systems*, 11(4):284–295. (Cited on page 30).

Chabini, I. and Lan, S. (2002a). Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *Intelligent Transportation Systems, IEEE Transactions on*, 3(1):60–74. (Cited on page 13).

Chabini, I. and Lan, S. (2002b). Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *Intelligent Transportation Systems, IEEE Transactions on*, 3(1):60–74. (Cited on page 33).

Chaturvedi, A., Filatyev, S., Gore, J., Hanna, A., Means, J., and Mellema, A. (2005). Integrating fire, structure and agent models. *Computational Science–ICCS 2005*, pages 695–702. (Cited on page 4).

Chen, S.-M. and Chien, C.-Y. (2011). Parallelized genetic ant colony systems for solving the traveling salesman problem. *Expert Systems with Applications*, 38(4):3873–3883. (Cited on page 34).

Cheng, T. (1999). *A process-oriented data model for fuzzy spatial objects*. PhD thesis, Wageningen Aricultural University. (Cited on page 22).

Chitumalla, P. K., Harris, D., Thuraisingham, B., and Khan, L. (2008). Emergency response applications: Dynamic plume modeling and real-time routing. *IEEE Internet Computing*, 12(1):38–44. (Cited on page 16).

Collier, N. (2003). Repast: An extensible framework for agent simulation. *The University of Chicagoos Social Science Research*, 36(February 11). (Cited on page 9).

Darema, F. (2004). Dynamic data driven applications systems: A new paradigm for application simulations and measurements. *Computational Science-ICCS 2004*, pages 662–669. (Cited on pages 4, 15, and 20).

De Almeida, V. and Güting, R. (2005). Indexing the trajectories of moving objects in networks. *GeoInformatica*, 9(1):33–60. (Cited on page 28).

de Bruijn, P. and Wijngaards, N. (2013). Agent-enabled information provisioning while retaining control: a demonstration. In *Intelligent Systems for Crisis Management*, pages 341–347. Springer. (Cited on page 29).

Delling, D., Sanders, P., Schultes, D., and Wagner, D. (2009). Engineering route planning algorithms. In *Algorithmics of large and complex networks*, pages 117–139. Springer. (Cited on page 13).

Dias, M. B., Zlot, R., Kalra, N., and Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270. (Cited on page 35).

Diehl, S., Neuvel, J., Zlatanova, S., and Scholten, H. (2006). Investigation of user requirements in the emergency response sector: the Dutch case. *Proceedings of the second Gi4DM*, pages 25–26. (Cited on pages 1 and 58).

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271. (Cited on page 32).

Dilo, A. (2006). *Representation of and reasoning with vagueness in spatial information: a system for handling vague objects*. PhD thesis, Wageningen University & ITC. (Cited on page 22).

Dilo, A. and Zlatanova, S. (2008). Spatiotemporal data modeling for disaster management in the Netherlands. *Information Systems for Crisis Response and Management*, pages 517–528. (Cited on pages 19 and 25).

Dilo, A. and Zlatanova, S. (2011). A data model for operational and situational information in emergency response. *Applied Geomatics*, 3(4):207–218. (Cited on pages 24, 25, 27, 54, 124, and 161).

Dorigo, M. and Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 1(1):53–66. (Cited on page 34).

Douglas, C., Beezley, J., Coen, J., Li, D., Li, W., Mandel, A., Mandel, J., Qin, G., and Vodacek, A. (2006). Demonstrating the validity of a wildfire DDDAS. In Alexandrov, V., van Albada, G., Sloot, P., and Dongarra, J., editors, *Computational Science – ICCS 2006*, volume 3993 of *Lecture Notes in Computer Science*, pages 522–529. Springer Berlin Heidelberg. (Cited on page 20).

Duchêne, C., Ruas, A., and Cambier, C. (2012). The CartACom model: transforming cartographic features into communicating agents for cartographic generalisation. *International Journal of Geographical Information Science*, 26(9):1533–1562. (Cited on page 30).

Duives, D. C., Daamen, W., and Hoogendoorn, S. P. (2013). State-of-the-art crowd motion simulation models. *Transportation Research Part C: Emerging Technologies*, 37:193–209. (Cited on page 3).

Ehlen, M. A., Sun, A. C., Pepple, M. A., Eidson, E. D., and Jones, B. S. (2014). Chemical supply chain modeling for analysis of homeland security events. *Computers & Chemical Engineering*, 60:102–111. (Cited on page 29).

El-Korany, A. and El-Bahnasy, K. (2008). A multi-agent cooperative model for crisis management system. In *7th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES (AIKED'08), University of Cambridge, UK*. (Cited on page 32).

Epstein, J., Pankajakshan, R., and Hammond, R. (2011). Combining computational fluid dynamics and agent-based modeling: A new approach to evacuation planning. *PloS one*, 6(5):e20139. (Cited on pages 31 and 162).

Fan, Z. and Zlatanova, S. (2011). Exploring ontologies for semantic interoperability of data in emergency response. *Applied Geomatics*, 3(2):109–122. (Cited on page 17).

Fiedrich, F. and Burghardt, P. (2007). Agent-based systems for disaster management. *Communications of the ACM*, 50(3):41–42. (Cited on page 29).

Filippoupolitis, A. and Gelenbe, E. (2009). A decision support system for disaster management in buildings. In *Proceedings of the Summer Computer Simulation Conference*, pages 141–147. Society for Modeling & Simulation International. (Cited on page 102).

Filippoupolitis, A., Gorbil, G., and Gelenbe, E. (2011). Autonomous navigation systems for emergency management in buildings. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 1056–1061. IEEE. (Cited on page 29).

Filippoupolitis, A., Loukas, G., Timotheou, S., Dimakis, N., and Gelenbe, E. (2009). Emergency response systems for disaster management in buildings. In *Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management, Bucharest, Romania*, pages 1–14. (Cited on page 102).

Ge, S. S. and Cui, Y. (2002). Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3):207–222. (Cited on page 13).

Genc, Z., Heidari, F., Oey, M. A., van Splunter, S., and Brazier, F. M. (2013a). Agent-based information infrastructure for disaster management. In *Intelligent Systems for Crisis Management*, pages 349–355. Springer. (Cited on page 29).

Genc, Z., Heidari, F., Oey, M. A., van Splunter, S., and Brazier, F. M. (2013b). Agent-based information infrastructure for disaster management. In *Intelligent Systems for Crisis Management*, pages 349–355. Springer. (Cited on page 31).

George, B., Kim, S., and Shekhar, S. (2007). Spatio-temporal network databases and routing algorithms: A summary of results. In *Advances in Spatial and Temporal Databases*, pages 460–477. Springer. (Cited on page 33).

Girard, G., Côté, S., Zlatanova, S., Barette, Y., St-Pierre, J., and Van Oosterom, P. (2011). Indoor pedestrian navigation using foot-mounted IMU and portable ultrasound range sensors. *Sensors*, 11(8):7606–7624. (Cited on page 13).

Glemser, M. and Klein, U. (2000). Hybrid modelling and analysis of uncertain data. *International Archive of Photogrammetry and Remote sensing, Amsterdam*. (Cited on pages 22, 23, and 161).

Gonzalez, J. P., Dornbush, A., and Likhachev, M. (2012a). Using state dominance for path planning in dynamic environments with moving obstacles. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4009–4015. IEEE. (Cited on page 90).

Gonzalez, J. P., Phillips, M., Neuman, B., and Likhachev, M. (2012b). Robust mobility in human-populated environments. In *Proc. SPIE*, volume 8387, pages 83870Z–1–83870Z–14. (Cited on page 16).

Gonzalez, J. P. and Stentz, A. (2005). Planning with uncertainty in position an optimal and efficient planner. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2435–2442. IEEE. (Cited on page 90).

Greenwood, D., Sharma, S., and Johansson, A. (2014). Gap analysis of current industrial challenges and the state-of-the-art in pedestrian modelling. *Transportation Research Procedia*, 2:219–227. (Cited on page 3).

Gunawan, L. T. (2013). *Crowdsourced Disaster Response for Effective Mapping and Wayfinding*. PhD thesis, Delft University of Technology. (Cited on page 14).

Güting, R., De Almeida, V., and Ding, Z. (2006). Modeling and querying moving objects in networks. *The VLDB Journal*, 15(2):165–190. (Cited on page 28).

Güting, R. H., Böhlen, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M., and Vazirgiannis, M. (2000). A foundation for representing and querying moving objects. *ACM Transactions on Database Systems (TODS)*, 25(1):1–42. (Cited on page 64).

Güting, R. H. and Schneider, M. (2005). *Moving objects databases*. Morgan Kaufmann. (Cited on page 64).

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107. (Cited on pages 32 and 80).

Holand, J. (1992). Adaptation in nature and artificial systems. (Cited on page 34).

Hoogendoorn, S., Landman, R., van Kooten, J., Taale, H., and Schreuder, M. (2014). Integrated network management amsterdam: Towards a field operational test. In *Transportation Research Board 93rd Annual Meeting*, number 14-2755. (Cited on page 3).

Hu, X. (2011). Dynamic Data Driven Simulation. *SCS M& S Magazine*, pages 662–669. (Cited on pages 4, 20, 21, and 161).

Huang, B., Wu, Q., and Zhan, F. (2007). A shortest path algorithm with novel heuristics for dynamic transportation networks. *International Journal of Geographical Information Science*, 21(6):625–644. (Cited on page 13).

Ibri, S., Nourelfath, M., and Drias, H. (2012). A multi-agent approach for integrated emergency vehicle dispatching and covering problem. *Engineering Applications of Artificial Intelligence*, 25(3):554–565. (Cited on page 30).

Jiang, N. and Parashar, M. (2009). Enabling end-to-end data-driven sensor-based scientific and engineering applications. In Allen, G., Nabrzyski, J., Seidel, E., van Albada, G., Dongarra, J., and Sloot, P., editors, *Computational Science – ICCS 2009*, volume 5545 of *Lecture Notes in Computer Science*, pages 449–459. Springer Berlin Heidelberg. (Cited on page 20).

Johnson, R. (2008). GIS technology and applications for the fire services. In Zlatanova, S. and Li, J., editors, *Geospatial Information Technology for Emergency Response*, Geospatial Information Technology for Emergency Response, pages 351–372. Taylor & Francis Ltd, London. (Cited on page 14).

Khoury, H. M. and Kamat, V. R. (2009). Evaluation of position tracking technologies for user localization in indoor construction environments. *Automation in Construction*, 18(4):444–457. (Cited on page 13).

Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., and Shimada, S. (1999). Robocup rescue: search and rescue in large-scale disasters as a domain for autonomous agents research. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 739–743 vol.6. (Cited on page 29).

Knoop, V. L. and Hoogendoorn, S. P. (2014). Network transmission model: a dynamic traffic model at network level. In *Transportation Research Board 93rd Annual Meeting*, number 14-1104. (Cited on page 3).

Koenig, S., Tovey, C., Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., Kleywegt, A., Meyerson, A., and Jain, S. (2006). The power of sequential single-item auctions for agent coordination. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1625. (Cited on pages 36, 98, and 164).

Krüger, C. and Lakes, T. (2015). Bayesian belief networks as a versatile method for assessing uncertainty in land-change modeling. *International Journal of Geographical Information Science*, 29(1):111–131. (Cited on page 21).

Kwan, M.-P. and Lee, J. (2005). Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments. *Computers, Environment and Urban Systems*, 29(2):93–113. (Cited on pages 25 and 26).

Lagoudakis, M. G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A. J., Koenig, S., Tovey, C. A., Meyerson, A., and Jain, S. (2005). Auction-based multi-robot routing. In *Robotics: Science and Systems*, volume 5. (Cited on page 98).

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., and Weibel, R. (1999). The application of agents in automated map generalisation. In *19th international cartographic conference*, volume 2, pages 1225–1234. (Cited on page 30).

Lee, L., Ndumu, D., and Nwana, H. (1998). Zeus: An advanced tool-kit for engineering distributed multi-agent systems. *the Proceeding of the practical application of Intelligent Agents and Multi-Agent Systems*, pages 377–392. (Cited on page 9).

Li, H., Yang, S. X., and Seto, M. L. (2009). Neural-network-based path planning for a multirobot system with moving obstacles. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(4):410–419. (Cited on page 16).

Li, N. and Becerik-Gerber, B. (2011). Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment. *Advanced Engineering Informatics*, 25(3):535–546. (Cited on page 13).

Liu, Y., Hatayama, M., and Okada, N. (2006). Development of an adaptive evacuation route algorithm under flood disaster. *Annuals of Disaster Prevention Research Institute, Kyoto University*, 49:189–195. (Cited on page 33).

López, T., Brintrup, A., McFarlane, D., and Dwyer, D. (2010). Selecting a multi-agent system development tool for industrial applications: A case study of self-serving aircraft assets. In *4th IEEE International Conference on Digital Ecosystems and Technologies-Conference Proceedings of IEEE-DEST 2010, DEST 2010*, pages 400–405. (Cited on page 9).

Lu, G., Wu, Z., Wen, L., Lin, C. A., Zhang, J., and Yang, Y. (2008). Real-time flood forecast and flood alert map over the Huaihe river basin in China using a coupled hydro-meteorological modeling system. *Science in China Series E: Technological Sciences*, 51(7):1049–1063. (Cited on page 15).

Lucor, D., Su, C.-H., and Karniadakis, G. E. (2004). Karhunen-Loeve representation of periodic second-order autoregressive processes. In *Computational Science-ICCS 2004*, pages 827–834. Springer. (Cited on page 20).

Luke, S., Balan, G., Panait, L., Cioffi-Revilla, C., and Paus, S. (2003). MASON: A Java multi-agent simulation library. In *Proceedings of Agent 2003 Conference on Challenges in Social Simulation*, volume 9. (Cited on page 9).

Luke, S., Cioffi-Revilla, C., Panait, L., and Sullivan, K. (2004). Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 SwarmFest Workshop*, volume 8. (Cited on page 102).

Lukosch, H., van Nuland, B., van Ruijven, T., van Veen, L., and Verbraeck, A. (2014). Building a virtual world for team work improvement. In *Frontiers in Gaming Simulation*, pages 60–68. Springer. (Cited on page 3).

Lukosch, H., van Ruijven, T., and Verbraeck, A. (2012). The other city–designing a serious game for crisis training in close protection. In *ISCRAM 2012 Conference Proceedings*. (Cited on page 3).

Mandel, J., Beezley, J., Bennethum, L., Chakraborty, S., Coen, J., Douglas, C., Hatcher, J., Kim, M., and Vodacek, A. (2007). A dynamic data driven wildland fire model. *Lecture Notes in Computer Science*, 4487:1042–1049. (Cited on page 4).

Masehian, E. and Katebi, Y. (2007). Robot motion planning in dynamic environments with moving obstacles and target. *International Journal of Mechanical Systems Science and Engineering*, 1(1):20–25. (Cited on pages 16, 17, and 161).

Meijers, M. (2011). *Variable-scale Geo-information*. PhD thesis, Delft University of Technology. (Cited on page 130).

Meijers, M., Zlatanova, S., and Pfeifer, N. (2005). 3D Geo-information indoors: structuring for evacuation. In *Proceedings of Next generation 3D city models*, pages 21–22. (Cited on page 24).

Meratnia, N. (2005). *Towards database support for moving object data*. PhD thesis, University of Twente. (Cited on page 28).

Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The swarm simulation system: A toolkit for building multi-agent simulations. *Technical report*. (Cited on page 9).

Mioc, D., Anton, F., and Liang, G. (2008). On-line street network analysis for flood evacuation planning. In *Remote Sensing and GIS Technologies for Monitoring and Prediction of Disasters*, pages 219–242. Springer. (Cited on pages 15, 16, 25, and 161).

Mioc, D., Moreiri, K., Nkhwanana, J., Nikerson, B., McGillivray, E., Morton, A., and Tang, P. (2012). *On-line early warning system for evacuation of socially vulnerable population during flooding*. WIT Transactions on Ecology and the Environment. WIT Press. (Cited on page 16).

Moreno, A., Posada, J., Segura, Á., Arbelaiz, A., and García-Alonso, A. (2014). Interactive fire spread simulations with extinguishment support for virtual reality training tools. *Fire Safety Journal*, 64:48–60. (Cited on page 15).

Moreno, A., Segura, Á., Korchi, A., Posada, J., and Otaegui, O. (2011). Interactive urban and forest fire simulation with extinguishment support. In *Advances in 3D Geo-Information Sciences*, pages 131–148. Springer. (Cited on pages 20, 21, 48, and 103).

Nanjanath, M. and Gini, M. (2006). Dynamic task allocation for robots via auctions. In *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, pages 2781–2786. IEEE. (Cited on pages 35, 36, and 162).

Nannicini, G., Delling, D., Schultes, D., and Liberti, L. (2012). Bidirectional A* search on time-dependent road networks. *Networks*, 59(2):240–251. (Cited on page 33).

Narayanan, V., Phillips, M., and Likhachev, M. (2012). Anytime safe interval path planning for dynamic environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4708–4715. (Cited on pages 84 and 88).

Nedkov, S. and Zlatanova, S. (2011). Enabling obstacle avoidance for Google maps' navigation service. In *Altan, Backhause, Boccardo & Zlatanova (Eds.), International Archives ISPRS XXXVIII, 7th, Gi4DM*, Anltalya, Turkey. (Cited on pages 14, 15, and 161).

Ni, J. and Yang, S. X. (2011). Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments. *IEEE Transactions on Neural Networks*, 22(12):2062–2077. (Cited on page 16).

Nourjou, R., Hatayama, M., Smith, S. F., Sadeghi, A., and Szekely, P. (2013). Design of a GIS-based assistant software agent for the incident commander to coordinate emergency response operations. *In Workshop on Robots and Sensors integration in future rescue INformation system (ROSIN' 13)*. (Cited on page 32).

Parker, C. J., MacFarlane, R., and Phillips, C. (2008). Integrated emergency management: Experiences and challenges of a national geospatial information provider, ordnance survey. In Zlatanova, S. and Li, J., editors, *Geospatial*

*Information Technology for Emergency Response*, volume 6 of *Geospatial Information Technology for Emergency Response*, pages 275–310. Taylor & Francis, London. (Cited on page 14).

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77. (Cited on page 6).

Penninga, F. and van Oosterom, P. (2008). A simplicial complex-based DBMS approach to 3D topographic data modelling. *International Journal of Geographical Information Science*, 22(7):751–779. (Cited on page 24).

Phillips, M. and Likhachev, M. (2011a). Planning in domains with cost function dependent actions. In *in Proceedings of AAAI Conference on Artificial Intelligence*. (Cited on page 33).

Phillips, M. and Likhachev, M. (2011b). SIPP: Safe interval path planning for dynamic environments. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. (Cited on pages 16, 33, 34, 84, 88, and 162).

Pokahr, A., Braubach, L., and Lamersdorf, W. (2005). Jadex: A BDI reasoning engine. In *Multi-agent programming*, pages 149–174. Springer. (Cited on page 9).

Poslad, S., Buckle, P., and Hadingham, R. (2000). The FIPA-OS agent platform: Open source for open standards. In *Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, volume 355, page 368. (Cited on page 9).

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002. (Cited on page 35).

Rajendran, B. (2009). 3D agent based model of cell growth. Master's thesis, University of Cincinnati. (Cited on page 9).

Rodríguez, R., Cortés, A., and Margalef, T. (2009). Injecting dynamic realtime data into a DDDAS for forest fire behavior prediction. *Computational Science–ICCS 2009*, pages 489–499. (Cited on pages 4, 15, and 20).

Rothkrantz, L. J. (2009). Dynamic routing using the network of car drivers. In *Proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to increase Digital Citizenship*, page 11. ACM. (Cited on page 102).

Rufli, M., Ferguson, D., and Siegwart, R. (2009). Smooth path planning in constrained environments. In *IEEE International Conference on Robotics and Automation, 2009. ICRA'09.*, pages 3780–3785. IEEE. (Cited on page 33).

Saadatmand-Tarzjan, M., Khademi, M., Akbarzadeh-T, M.-R., and Moghaddam, H. A. (2007). A novel constructive-optimizer neural network for the traveling salesman problem. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(4):754–770. (Cited on page 34).

Šaltenis, S., Jensen, C., Leutenegger, S., and Lopez, M. (2000). Indexing the positions of continuously moving objects. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, volume 29, pages 331–342. ACM. (Cited on page 28).

Schmitz, S., Zipf, A., and Neis, P. (2008). New applications based on collaborative geodata—the case of routing. In *Proceedings of XXVIII INCA International Congress on Collaborative Mapping and Space Technology*. (Cited on page 15).

Schoenharl, T. and Madey, G. (2011). Design and implementation of an agent-based simulation for emergency response and crisis management. *Journal of Algorithms* & *Computational Technology*, 5(4):601–622. (Cited on page 28).

Schoenharl, T., Madey, G., Szabó, G., and Barabási, A. (2006). WIPER: A multi-agent system for emergency response. In *Proceedings of the 3rd International ISCRAM Conference*. (Cited on page 30).

Schoenig, A. and Pagnucco, M. (2011). Evaluating sequential single-item auctions for dynamic task allocation. In *AI 2010: Advances in Artificial Intelligence*, pages 506–515. Springer. (Cited on pages 36 and 98).

Schurr, N., Marecki, J., Lewis, J. P., Tambe, M., and Scerri, P. (2005). The defacto system: Coordinating human-agent teams for the future of disaster response. In *Multi-Agent Programming*, pages 197–215. Springer. (Cited on pages 28 and 29).

Sengupta, R. and Sieber, R. (2007). Geospatial agents, agents everywhere... *Transactions in GIS*, 11(4):483–506. (Cited on page 30).

Shastri, J. (2006). Safe navigation during fire hazards using specknets. Master's thesis, The University of Edinburgh, Edinburgh, UK. (Cited on pages 103 and 104).

Sheremetov, L. B., Contreras, M., and Valencia, C. (2004). Intelligent multi-agent support for the contingency management system. *Expert Systems with Applications*, 26(1):57–71. (Cited on page 30).

Shi, W. (2008). Modeling uncertainty in geographic information and analysis. *Science in China Series E: Technological Sciences*, 51(1):38–47. (Cited on page 21).

Shi, W., Cheung, C.-K., and Tong, X. (2004). Modelling error propagation in vector-based overlay analysis. *ISPRS journal of photogrammetry and remote sensing*, 59(1):47–59. (Cited on page 23).

Sistla, A. P., Wolfson, O., Chamberlain, S., and Dao, S. (1997). Modeling and querying moving objects. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 422–422. IEEE Computer Society. (Cited on page 28).

Skiena, S. S. (2008). *The Algorithm Design Manual*. Springer. (Cited on pages 97 and 164).

Sonti, S., Virani, N., Jha, D. K., Mukherjee, K., and Ray, A. (2013). Language measure-theoretic path planning in the presence of dynamic obstacles. In *American Control Conference (ACC), 2013*, pages 5110–5115. IEEE. (Cited on page 16).

Speičvcys, L., Jensen, C., and Kligys, A. (2003). Computational data modeling for network-constrained moving objects. In *Proceedings of the 11th ACM international symposium on Advances in geographic information systems*, pages 118–125. ACM. (Cited on pages 28 and 161).

Stoter, J. and van Oosterom, P. (2002). Incorporating 3D geo-objects into a 2D geo-DBMS. In *Proceedings of ASPRS/ACSM*, pages 19–26. (Cited on page 24).

Stoter, J. and Zlatanova, S. (2003). 3D GIS, where are we standing. In *Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modeling and Analysis*. (Cited on page 24).

Sullivan, K., Coletti, M., and Luke, S. (2010). Geomason: Geospatial support for mason. *Department of Computer Science, George Mason University, Technical Report Series*. (Cited on page 102).

Tang, F. and Zhang, X. (2008). A GIS-Based 3D simulation for occupant evacuation in a building. *Tsinghua Science & Technology*, 13:58–64. (Cited on page 31).

Teodorović, D. and Pavković, G. (1996). The fuzzy set theory approach to the vehicle routing problem when demand at nodes is uncertain. *Fuzzy sets and systems*, 82(3):307–317. (Cited on pages 29 and 35).

Trafalis, T. B., Santosa, B., and Richman, M. B. (2004). Rule-based support vector machine classifiers applied to tornado prediction. In *Computational Science-ICCS 2004*, pages 678–684. Springer. (Cited on page 20).

Van Borkulo, E., Scholten, H. J., Zlatanova, S., and van den Brink, A. (2005). Decision making in response and relief phases. *Geo-information for disaster management-late papers*, pages 47–54. (Cited on page 19).

Van Den Berg, J. P. and Overmars, M. H. (2005). Roadmap-based motion planning in dynamic environments. *IEEE Transactions on Robotics*, 21(5):885–897. (Cited on pages 33 and 91).

van Oosterom, P. (2009). Research and development in geo-information generalisation and multiple representation. *Computers, Environment and Urban Systems*, 33(5):303–310. (Cited on page 25).

van Oosterom, P. and Lemmen, C. (2002). Impact analysis of recent geo-ICT developments on cadastral systems. In *FIG XXII Congres, Washington DC, USA*. (Cited on page 24).

van Oosterom, P., Vertegaal, W., Van Hekken, M., and Vijlbrief, T. (1994). Integrated 3D modelling within a GIS. In *AGDM*, pages 80–95. (Cited on page 24).

Verbree, E., Zlatanova, S., van Winden, K., van der Laan, E., Makri, A., Taizhou, L., and Haojun, A. (2013). To localise or to be localised with WiFi in the Hubei museum? *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(4):31–35. (Cited on page 13).

Visser, I. (2009). Route determination in disaster areas. Master's thesis, GIMA MSc programme (Utrecht University, TU Delft, Wageningen University, ITC). (Cited on page 33).

Vokřínek, J., Komenda, A., and Pěchouček, M. (2010). Agents towards vehicle routing problems. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 773–780. (Cited on pages 29 and 35).

Wang, Z. and Zlatanova, S. (2013a). An A*-based search approach for navigation among moving obstacles. In *Intelligent Systems for Crisis Management*, pages 17–30. Springer. (Cited on page 79).

Wang, Z. and Zlatanova, S. (2013b). Multi-agent infrastructure assisting navigation for first responders. In *Proceedings of the Sixth ACM SIGSPATIAL*

*International Workshop on Computational Transportation Science*, IWCTS '13, pages 1–6, New York, NY, USA. ACM. (Cited on pages 13 and 69).

Wang, Z. and Zlatanova, S. (2013c). Taxonomy of navigation for first responders. In *Progress in Location-Based Services*, pages 297–315. Springer. (Cited on pages 13, 14, 37, 129, 177, and 181).

Wang, Z. and Zlatanova, S. (2014). Multi-agent based path planning for first responders among moving obstacles. *Computers, Environment and Urban Systems*. under review. (Cited on pages 51, 69, and 79).

Wang, Z., Zlatanova, S., Moreno, A., van Oosterom, P., and Toro, C. (2014). A data model for route planning in the case of forest fires. *Computers & Geosciences*, 68:1–10. (Cited on pages 13, 51, and 79).

Wang, Z., Zlatanova, S., and van Oosterom, P. (2015). Path planning for first responders among uncertain moving obstacles. Manuscript to be submitted to an international scientific journal. (Cited on pages 13 and 51).

Wilensky, U. (1999). Netlogo: Center for connected learning and computer-based modeling. *Northwestern University, Evanston, IL*, pages 49–52. (Cited on page 9).

Wolfson, O., Xu, B., Chamberlain, S., and Jiang, L. (1998). Moving objects databases: Issues and solutions. In *Proceedings of the 10th International Conference on Scientific and Statistical Database Management*, pages 111–122. IEEE Computer Society. (Cited on page 28).

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice. *The knowledge engineering review*, 10(02):115–152. (Cited on page 28).

Wulterkens, D. (2007). EMS in The Netherlands: A Dutch Treat? *Journal of Emergency Medical Services. Tuesday, Nov 27th*. (Cited on page 59).

Xu, W., Dilo, A., Zlatanova, S., and van Oosterom, P. (2008). Modelling emergency response processes: comparative study on OWL and UML. *Information systems for crisis response and management, Harbin Engineering University*, pages 493–504. (Cited on page 17).

Yang, S. X., Hu, Y., and Meng, M. Q. H. (2006). A knowledge based ga for path planning of multiple mobile robots in dynamic environments. In *2006 IEEE Conference on Robotics, Automation and Mechatronics*, 2006 IEEE Conference on Robotics, Automation and Mechatronics, pages 1–6, Bangkok. (Cited on page 16).

Zarboutis, N. and Marmaras, N. (2007). Design of formative evacuation plans using agent-based simulation. *Safety Science*, 45(9):920–940. (Cited on page 29).

Zelle, H., Wisse, E., Mika, Á., and van Tilburg, T. (2013). Smoke plume modeling in crisis management. In *Intelligent Systems for Crisis Management*, pages 51–66. Springer. (Cited on pages 3, 15, 20, 48, and 161).

Zheng, X., Koenig, S., and Tovey, C. (2006). Improving sequential single-item auctions. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2238–2244. IEEE. (Cited on page 36).

Zlatanova, S. (2000). *3D GIS for urban development*. PhD thesis, ITC, The Netherlands. (Cited on page 24).

Zlatanova, S. (2010). Formal modeling of tasks to support search of geo-information in emergency response. In *International symposium climate change and disaster management. GIS Development Pvt. Ltd., Noida*, pages 26–36. (Cited on pages 1, 17, 18, 58, and 161).

Zlatanova, S. and Baharin, S. S. K. (2008). Optimal navigation of first responders using DBMS. In *Joint conference of the 3rd international conference on information systems for crisis response and management/4th international symposium on geo-information for disaster management*, pages 541–554. (Cited on pages 25, 27, and 38).

Zlatanova, S., Dilo, A., Vries, M. d., and Fichtinger, A. (2010). Models of dynamic data for emergency response: a comparative study. In *Proceedings special joint symposium of ISPRS Technical Commission IV & AutoCarto in conjunction with ASPRS/CaGIS*. American Society for Photogrammetry and Remote Sensing, International Society for Photogrammetry and Remote Sensing, and Cartography and Geographic Information Society. (Cited on pages 25, 26, 27, and 161).

Zlatanova, S. and Holweg, D. (2004). 3D Geo-information in emergency response: a framework. In *Proceedings of the 4th International Symposium on Mobile Mapping Technology (MMT'2004), March*, pages 29–31. (Cited on page 25).

# APPENDIX A

In this appendix, we first give a short introduction of the general framework of the developed client-server navigation system (see Section A.1). In Section A.2, we present the developments at the server side. Section A.3 describes in detail the client application, which can use the services provided by the server.

## A.1 The general framework

The general framework of the client-server navigation system we developed is shown in Figure A.1. Basically it consists of two parts: 1). The server, which is built using the agent simulation tool. It conducts the spatial data processing to extract the information essential for routing, according to the data models presented in Chapter 4, and performs the route calculation, using the algorithms presented in Chapter 6; 2). The client application, which allows users to configure the request that is sent to the server, and supports visualizing the routes and obstacles, which are provided by the server.

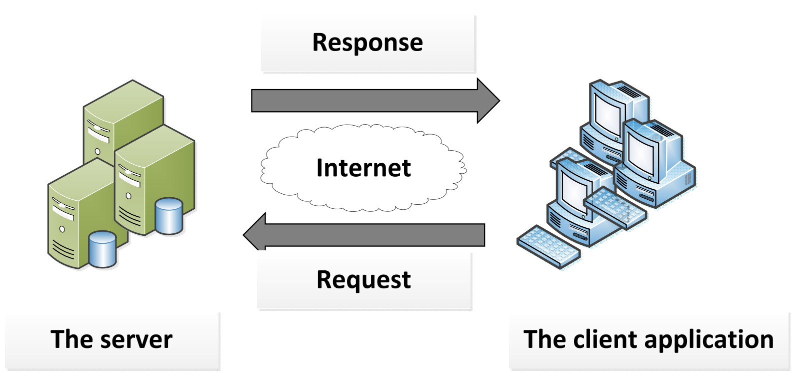



Figure A.1: General framework

### A.2 The server side

The server provides two services for the client application: 1). Obstacle delivery service; 2). Routing service. HTTP protocol (Hypertext Transfer Protocol) is used to link the server and the client application. On the server side, we use classes, ObstacleServer and RoutingServer, which extend HttpServlet, to handle the request from the client application. Classes RouteData and ObstacleData are used to configure the request and to store the data sent from the server.

#### A.2.1 Obstacles delivery service

*1. Class overview*

Package server

- Class ObstacleServer extends HttpServlet
  - protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    Called by the server to allow a servlet to handle the request from the client side
- Class ObstacleData
  - public void set_sql(String sql)
    This method sets the sql command that is sent to the server to query for the data of moving obstacles. Here we use the following sql statement to retrieve the data obstacles from the PostgreSQL

```
SELECT row_to_json(fc)
FROM ( SELECT 'FeatureCollection' As type,
    array_to_json(array_agg(f)) As features
FROM (SELECT 'Feature' As type
   , ST_AsGeoJSON(lg.geog)::json As geometry
   , row_to_json((SELECT l FROM (SELECT loc_id, loc_name) As l
     )) As properties
  FROM locations As lg ) As f ) As fc;
```

  - public String get_sql()
    This method gets the sql statement.
  - public void set_obstacles_str (String obstacles_str)
    This method sets the GeoJSON string of the obstacles obtained from the PostgreSQL

- public String get_obstacles_str()
  This method gets the GeoJSON string of the obstacles

*2. Description*

After an obstacle delivery request is received from the client application, the ObstacleSever will call getInputStream() and cast the object from InputStream to an instance of ObstacleData. ObstacleSever will also call the method get_sql() to get the sql command and run the query to retrieve the data of obstacles. When the data retrieving is done, the data of obstacles will be set in the ObstacleData as a GeoJSON string and sent to the client application in the OutputStream.

### A.2.2 Routing service

*1. Class overview*

Package server

- Class RoutingServer extends HttpServlet

  - protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    Called by the server to allow a servlet to handle the request from the client side

- Class RouteData

  Field summary

  - sourceLon
    the longitude value of the source
  - sourceLat
    the latitude value of the source
  - destinationLon
    the longitude value of the destination
  - destinationLat
    the latitude value of the destination

  Method summary

  - public void setOptimalRoute(String optimalRouteString)
    This method sets the optimal route string
  - public void setShortestRoute(String shortestRouteString)
    This method sets the shortest route string

- public String getShortestRoute( )
  This method gets the shortest route string
- public String getOptimalRoute()
  This method gets the optimal route string
- public void setDepartureTimeStr(String departureTimeStr)
  This method sets the departure time in the string format
- public String getDepartureTimeStr()
  This method gets the departure time in the string format
- public void setSpeed(double speed)
  This method sets the speed of the vehicle
- public double getSpeed()
  This method sets the speed of the vehicle

*2. Description*

After a routing request is received from the client application, the RoutingSever will call getInputStream() and cast the object from InputStream to an instance of RouteData. When the calculation is done, the route string will be set in the RouteData and sent to the client application in the OutputStream.

### A.3 The client side

At the client side, we serialize the objects of RouteData and ObstacleData, and send them to the server via URLConnection, as shown below.

```
RouteData rd = new RouteData(5.8882,51.9465,5.9691,51.9712);
String departureTimeStr = "2013-02-16 22:40:15";
rd.setDepartureTimeStr(departureTimeStr);
rd.setSpeedFlag(+1);
rd.setSpeed(110);

URL url = new URL("http://emergency-otb.tudelft.nl/demo/router");
java.net.URLConnection con = url.openConnection();
con.setDefaultUseCaches(false);
con.setRequestProperty("Content-type",
    "application/x-java-serialized-object");
con.setUseCaches(false);
con.setDoOutput(true);
con.setDoInput(true);
con.connect();

OutputStream ops = con.getOutputStream();
ObjectOutputStream objout = new ObjectOutputStream(ops);
```

```
objout.writeObject(rd); /// send the object to the server
objout.flush();
objout.close();
```

Notes: Class RouteData at the server side and at the client side should be exactly the same. The same holds for the class ObstacleData.

### A.3.1 Obstacles delivery request

A obstacles delivery request takes the following form:
http://emergency-otb.tudelft.nl/demo/obstacles

*1. Class overview*

Package server

- Class ObstacleData
  This class is exactly the same as the class ObstacleData at the server side. It is used by both the server and the client application to store and transfer data of the moving obstacles.

*2. Output*

The output at the client side can be one of the following formats:
1). GeoJSON (recommended), the obstacle string returned from the sever.


```
{"type":"FeatureCollection",
"features":[{
"type":"Feature",
"geometry":{
"type":"Polygon",
"coordinates":[[[5.9084224685329,52.0249434960486],
[5.91401394016465,52.0246921939528],
[5.91640131007483,52.0228074282342],[5.91483067197603,52.0157709695516],
[5.90835964300895,52.0161479226953],[5.90521836681134,52.0180326884139],
[5.90383620528439,52.0217393943271],[5.9084224685329,52.0249434960486]]]},
"properties":{"gid":1,"time_id":1,"SOC":"BLUE","timestamp":"2013-02-16
    22:41:00"}
}]}
```


2). shapefile, which is transformed from GeoJSON and used by the agent simulator for visualization. Figure A.2 shows the table associated with obstacles.

The definition of attributes is as follows:

Attribute table - obstaces_s :: 0 / 11 feature(s) selected

| | gid | time_id | timestamp | SOC |
|---|---|---|---|---|
| 0 | 1 | 1 | 2013-02-16 2... | BLUE |
| 1 | 2 | 2 | 2013-02-16 2... | BLUE |
| 2 | 3 | 3 | 2013-02-16 2... | BLUE |
| 3 | 4 | 4 | 2013-02-16 2... | BLUE |
| 4 | 5 | 5 | 2013-02-16 2... | BLUE |
| 5 | 6 | 6 | 2013-02-16 2... | BLUE |
| 6 | 7 | 7 | 2013-02-16 2... | BLUE |
| 7 | 8 | 8 | 2013-02-16 2... | BLUE |
| 8 | 9 | 9 | 2013-02-16 2... | BLUE |
| 9 | 10 | 10 | 2013-02-16 2... | BLUE |
| 10 | 11 | 0 | 2013-02-16 2... | RED |

Figure A.2: Table of obstacles

- gid, the id of obstacle feature
- timestamp, the timestamp associated with the obstacle
- time_id, the time id corresponds to the timestamp and is used for simulation
- SOC, indicates if the obstacles is visualized by the simulator or not. RED means yes, BLUE means NO.

### A.3.2 Routing request

A routing request takes the following form:
http://emergency-otb.tudelft.nl/demo/router

*1. Class overview*

Package server

- Class RouteData
  This class is exactly the same as the class RouteData at the server side. It is used by both the server and the client application to store and transfer the information related to the routes.

*2. Request parameters*

Certain parameters are required while others are optional.

- Required parameters

  - source – the textual latitude/longitude value from which you wish to calculate routes
  - destination – The textual latitude/longitude value from which you wish to calculate routes

  All the values will be used to construct an instance of RouteData. Notes: the source and the destination must be located within the bounding box (as shown in Figure A.3) specified by the following coordinates: [(5.85847,52.03119), (5.98696,52.03665);(5.99320,51.93469) (5.86185,51.93079)]

- Optional parameters

  - speed, the traveling speed of the vehicle
  - departure time, the desired time of departure


Figure A.3: The navigable area

*3. Output*

The output will be in the following formats:

- GeoJSON, the route in this format is used by the agent simulator to configure the agent
- JSON, the route in this format is generated for the mobile device developed by NMPO[1]
- shapefile, the route in this format is used by agent simulator for visualization. It can also be visualized by traditional GIS tools, such as QGIS.

1). The GeoJSON result is shown below.


```
{"type":"FeatureCollection",
"features":[{
"type":"Feature",
"geometry":{
"type":"Feature","geometry":{"type":"LineString","coordinates":[[5.8735,52.0258],
[5.8727,52.0261],[5.8719,52.0263]]}
"properties":{"to_from_coord":"(5.96182, 52.00738, NaN)-(5.96187, 52.00731,
    NaN)"},
"type":"Feature","geometry":
{"type":"LineString","coordinates":[[5.9461,52.0105],[5.9454,52.0106],
[5.9447,552.02584]])","dir":"true","source_id":0,"length":121.688090309097,"road_name":
vanalvereneg","road_id":909340945,"waiting_time":0,"seq_in_route":80,"target_id":0}
}]}
```


The definition is as follows:

- road_id, the id of the road
- source_id, the source id of the road
- target_id, the target id of the road
- to_from_coord, indicate the direction of the road in the route
- dir, indicate if the direction of road is the same as the linestring of the road
- length, the length of the road
- road_name, the name of the road

[1] http://www.jewel.eu/

- seq_in_route, the sequence id of the road in the route
- waiting_time, the waiting time associated with the source node of this road

2). The JSON result is shown below.


```
[
   {
      "id": 1,
      "pid": 722268350,
      "lat": 52.00241,
      "lon": 5.65748,
      "road": {
         "st": "Van Balverenweg",
         "geom": "5.65748 52.00241,5.65769 52.002465,5.6579 52.00252",
         "l": 31,
         "h": 75
      },
      "tulip": "",
      "o": "",
      "i": "",
      "w": null,
      "ws": null,
      "actie": 1,
      "dosis": 0,
      "links": 2.66666666666667,
      "rechts": 1.33333333333333
   }, etc. etc. etc.]
```


The JSON string contains the following information:

- id is a sequence
- pid is the navteq parent_id
- lat and lon are the center points for a line
- road contains:
    - st = streetname
    - geom contains the full set of coordinates describing vector on which to drive
    - l = length in meters
    - h = heading. The angle of the road (North = 0 degrees)

- tulip contains a symbol reference for the tulip to present.

3). The route in shapefile and its table are shown in Figure A.4 and Figure A.5 respectively.

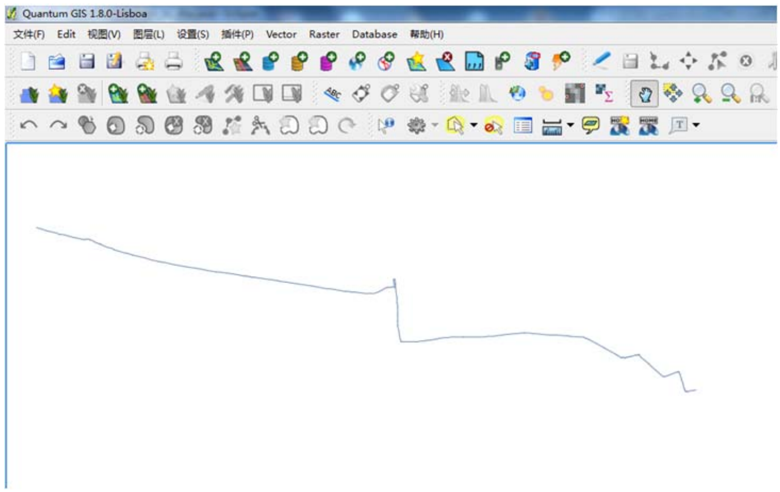



Figure A.4: Visualization of the route in QGIS

Attribute table - routes_s :: 0 / 81 feature(s) selected

| | to_from_co | dir | length | road_name | seq_in_rou | waiting_ti | source_id | road_id | target_id |
|---|---|---|---|---|---|---|---|---|---|
| 0 | (5.96913, 52.0... | false | 82.176539667... | van balvereneg | 0 | 0 | 0 | 832195752 | 0 |
| 1 | (5.96898, 52.0... | false | 10.828427690... | van balvereneg | 1 | 0 | 0 | 832195751 | 0 |
| 2 | (5.96877, 52.0... | true | 18.702252311... | van balvereneg | 2 | 0 | 0 | 547747038 | 0 |
| 3 | (5.96873, 52.0... | true | 14.723389947... | van balvereneg | 3 | 0 | 0 | 949362162 | 0 |
| 4 | (5.96864, 52.0... | true | 29.582751233... | van balvereneg | 4 | 0 | 0 | 949362163 | 0 |
| 5 | (5.96854, 52.0... | true | 35.170062540... | van balvereneg | 5 | 0 | 0 | 757954492 | 0 |
| 6 | (5.96841, 52.0... | true | 43.213959086... | van balvereneg | 6 | 0 | 0 | 756028212 | 0 |
| 7 | (5.96836, 52.0... | true | 17.039221728... | van balvereneg | 7 | 0 | 0 | 756028213 | 0 |
| 8 | (5.9682, 52.00... | true | 56.708308968... | van balvereneg | 8 | 0 | 0 | 756028214 | 0 |
| 9 | (5.96785, 52.0... | true | 113.85345292... | van balvereneg | 9 | 0 | 0 | 757954498 | 0 |
| 10 | (5.96775, 52.0... | true | 22.231210280... | van balvereneg | 10 | 0 | 0 | 757954499 | 0 |
| 11 | (5.96763, 52.0... | false | 9.9426687867... | van balvereneg | 11 | 0 | 0 | 949322852 | 0 |
| 12 | (5.96556, 52.0... | false | 163.60704189... | van balvereneg | 12 | 0 | 0 | 949322851 | 0 |
| 13 | (5.96374, 52.0... | true | 212.20798297... | van balvereneg | 13 | 0 | 0 | 68407393 | 0 |
| 14 | (5.96332, 52.0... | true | 49.358942933... | van balvereneg | 14 | 0 | 0 | 54877554 | 0 |
| 15 | (5.96316, 52.0... | true | 19.062252357... | van balvereneg | 15 | 0 | 0 | 54877465 | 0 |
| 16 | (5.96187, 52.0... | true | 159.37058620... | van balvereneg | 16 | 0 | 0 | 949322821 | 0 |
| 17 | (5.96182, 52.0... | true | 8.5122068794... | van balvereneg | 17 | 0 | 0 | 949322822 | 0 |
| 18 | (5.96159, 52.0... | false | 16.744798660... | van balvereneg | 18 | 0 | 0 | 756369576 | 0 |
| 19 | (5.96126, 52.0... | false | 24.345773300... | van balvereneg | 19 | 0 | 0 | 756369575 | 0 |
| 20 | (5.96071, 52.0... | false | 40.442159290... | van balvereneg | 20 | 0 | 0 | 755827762 | 0 |
| 21 | (5.96043, 52.0... | false | 20.744641199... | van balvereneg | 21 | 0 | 0 | 756021853 | 0 |
| 22 | (5.96002, 52.0 | false | 30.287307665 | van balvereneg | 22 | 0 | 0 | 756021852 | 0 |

Figure A.5: The table associated with the route

## LIST OF FIGURES

# LIST OF TABLES

## LIST OF ACRONYMS

| | |
|---|---|
| ABM | Agent-Based Modeling |
| ALGA | Artificial Life Geospatial Agent |
| CCC | command and control center |
| CDM | conceptual data model |
| DBMS | database management systems |
| DDDAS | Dynamic Data Driven Applications System |
| DM | disaster management |
| ER | emergency response |
| GIS | geographic information systems |
| geo-DBMS | geo-database management system |
| GRIP | Coordinated Regional Incident Suppression Procedure |
| ISO | International Standard Organization |
| HTTP | Hypertext Transfer Protocol |
| ITS | intelligent transportation system |
| JADE | Java Agent DEvelopment Framework |
| LBS | Location-Based Service |
| MAS | multi-agent system |
| MC | Monte Carlo |
| MOAAstar | Moving Obstacle Avoiding A* |

| | |
|---|---|
| MTSP | Multiple Travel Salesmen Problem |
| OGC | Open Geospatial Consortium |
| PE | pursuit-evasion |
| QGIS | Quantum GIS |
| SGA | Software Geospatial Agent |
| SPP | shortest path problem |
| SSI | Sequential Single-Item |
| SQL | Structured Query Language |
| TSP | Travel Salesmen Problem |
| UML | Unified Modeling Language |
| VGI | Volunteered Geographic Information |
| VRP | vehicle routing problem |

## LIST OF SYMBOLS

| | |
|---|---|
| $b_{\text{delay}}(T_j)$ | binary variable to indicate if the vehicle arrives later than the required arrival time of task $T_j$ |
| $CT(v, P)$ | the completion time of all tasks in path $P$ (in minutes) |
| $D$ | destination |
| $DN(v, P)$ | the number of delays of vehicle $v$ following path $P$, $= \sum_{j=1}^{k} b_{\text{delay}}(T_j)$ |
| $DT(v, P)$ | the sum of all time delays (in minutes), $= \sum_{j=1}^{k} t_{\text{delay}}(T_j)$ |
| $E$ | set of edges |
| $F(v, P)$ | the cost of the entire path $P$ of vehicle $v$ |
| $f(x)$ | the path cost associated with node $x$, $= g(x) + h(x)$ |
| $g(x)$ | the actual cost of the path from source $S$ to node $x$ |
| $G$ | a graph, $= (N, E)$ |
| $h(x)$ | the estimated cost from node $x$ to destination $D$ |
| $l_{xy}$ | the length of the edge $xy$ |
| $M_a$ | the number of times of being affected by the obstacles |
| $N_s$ | the number of simulation results generated by Monte Carlo method |
| $N$ | set of nodes |
| $P$ | path |
| $P_x$ | sequence of open intervals of node $x$ |
| $P_{xy}$ | sequence of open intervals of edge $xy$ |
| $P_b$ | the probability of being affected by the obstacles |
| $p_q^{x,\alpha}$ | the $q$-th full-safe ($\alpha$) interval of node $x$, = $(t_{oq}^{x,\alpha}, t_{cq}^{x,\alpha})$ |
| $p_m^{x,\beta}$ | the $m$-th partial-safe ($\beta$) interval of node $x$, = $(t_{om}^{x,\beta}, t_{cm}^{x,\beta})$ |
| $p_u^{x,\gamma}$ | the $u$-th non-safe ($\gamma$) interval of node $x$ |
| $p_q^{xy,\alpha}$ | the $q$-th full-safe interval ($\alpha$) of edge $xy$, = $(t_{oq}^{xy,\alpha}, t_{cq}^{xy,\alpha})$ |
| $p_m^{xy,\beta}$ | the $m$-th partial-safe interval ($\beta$) of edge $xy$, = $(t_{om}^{xy,\beta}, t_{cm}^{xy,\beta})$ |

| | |
|---|---|
| $p_u^{xy,\gamma}$ | the $u$-th non-safe interval ($\gamma$) of edge $xy$ |
| $p_k^x$ | the $k$-th open interval of node $x$, $= (t_{ok}^x, t_{ck}^x)$ |
| $p_j^{xy}$ | the $j$-th open interval of edge $xy$, $= (t_{oj}^{xy}, t_{cj}^{xy})$ |
| $R$ | a route |
| $r$ | risk variable representing the accumulated time of passing through the partially safe roads |
| $r_{max}$ | the maximal amount time that the responders are allowed to pass through partially safe roads |
| $\Delta r$ | the risk increment |
| $S$ | source |
| $S_x$ | set of states associated with node $x$ |
| $S_{n_i n_{i+1}}$ | the safety of edge $n_i n_{i+1}$ |
| $S_R$ | the safety of route $R$ |
| $s, s'$ | state |
| $T$ | set of tasks $\{T_1, T_2, \cdots, T_k\}$ |
| $\Delta t$ | timestep |
| $t_{\text{A}}(T_j)$ | the arrival time to the position of task $T_j$ |
| $t_{\text{D}}(T_j)$ | the departure time from the position of task $T_j$ |
| $t_{\text{O}}(T_j)$ | the operation time of task $T_j$ |
| $t_{\text{RA}}(T_j)$ | the required arrival time of task $T_j$ |
| $t_{\text{delay}}(T_j)$ | the time delay between $t_{\text{A}}(T_j)$ and $t_{\text{RA}}(T_j)$ of task $T_j$ |
| $t_w(s, s')$ | the waiting time needed for moving from state $s$ to state $s'$ |
| $t_{aw}$ | the additional waiting time |
| $t_{oq}^{x,\alpha}$ | the start time ($o$) of the $q$-th full-safe ($\alpha$) interval ($p_q^{x,\alpha}$) of node $x$ |
| $t_{cq}^{x,\alpha}$ | the end time ($c$) of the $q$-th full-safe ($\alpha$) interval ($p_q^{x,\alpha}$) of node $x$ |
| $t_{oq}^{xy,\alpha}$ | the start time ($o$) of the $q$-th full-safe ($\alpha$) interval ($p_q^{xy,\alpha}$) of edge $xy$ |
| $t_{cq}^{xy,\alpha}$ | the end time ($c$) of the $q$-th full-safe ($\alpha$) interval ($p_q^{xy,\alpha}$) of edge $xy$ |
| $t_{om}^{x,\beta}$ | the start time ($o$) of the $m$-th partial-safe ($\beta$) interval ($p_m^{x,\beta}$) of node $x$ |
| $t_{cm}^{x,\beta}$ | the end time ($c$) of the $m$-th partial-safe ($\beta$) interval ($p_m^{x,\beta}$) of node $x$ |
| $t_{om}^{xy,\beta}$ | the start time ($o$) of the $m$-th partial-safe ($\beta$) interval ($p_m^{xy,\beta}$) of edge $xy$ |
| $t_{cm}^{xy,\beta}$ | the end time ($c$) of the $m$-th partial-safe ($\beta$) interval ($p_m^{xy,\beta}$) of edge $xy$ |
| $t_{ok}^x$ | the start time ($o$) of the $k$-th open interval ($p_k^x$) of node $x$ |
| $t_{ck}^x$ | the end time ($c$) of the $k$-th open interval ($p_k^x$) of node $x$ |
| $t_{oj}^{xy}$ | the start time ($o$) of the $j$-th open interval ($p_j^{xy}$) of edge $xy$ |

| | |
|---|---|
| $t_{cj}^{xy}$ | the end time ($c$) of the $j$-th open interval ($p_j^{xy}$) of edge $xy$ |
| $t_c^{xy}$ | the time when the edge $xy$ is affected |
| $w_1, w_2$ | the penalty weights |
| $v$ | vehicle |
| $x$ | node |
| $y$ | node |
| $xy$ | the edge between node $x$ and node $y$ |

# SUMMARY


## Path planning for first responders in the presence of moving obstacles

Navigation services have gained much importance for all kinds of human activities ranging from tourist navigation to support of rescue teams in disaster management. However, despite the considerable amount of route guidance research that has been performed, many issues that are related to navigation for first responders still need to be addressed.

During disasters, emergencies can result in different types of moving obstacles (e.g., fires, plumes, floods), which make some parts of the road network temporarily unavailable. After such incidents occur, responders have to go to different destinations to perform their tasks in the environment affected by the disaster. Therefore they need a path planner that is capable of dealing with such moving obstacles, as well as generating and coordinating their routes quickly and efficiently.

During the past decades, more and more hazard simulations, which can modify the models with incorporation of dynamic data from the field, have been developed. These hazard simulations use methods such as data assimilation, stochastic estimation, and adaptive measurement techniques, and are able to generate more reliable results of hazards. This would allow the hazard simulation models to provide valuable information regarding the state of road networks affected by hazards, which supports path planning for first responders among the moving obstacles.

The objective of this research is to develop an integrated navigation system for first responders in the presence of moving obstacles. Such system should be able to navigate one or more responders to one or multiple destinations avoiding the moving obstacles, using the predicted information of the moving obstacles generated from by hazard simulations. In this dissertation, the objective we have is expressed as the following research question:

*How do we safely and efficiently navigate one or more first responders to one or more destinations avoiding moving obstacles?*

To address the above research questions, this research has been conducted using the following outline: 1). literature review; 2). conceptual design and analysis; 3). implementation of the prototype; and 4). assessment of the prototype and adaption. We investigated previous research related to navigation in disasters, and designed an integrated navigation system architecture, assisting responders in spatial data storage, processing and analysis. Within this architecture, we employ hazard models to provide the predicted information about the obstacles, and select a geo-database to store the data needed for emergency navigation. Throughout the development of the prototype navigation system, we have proposed:

- a taxonomy of navigation among obstacles, which categorizes navigation cases on basis of type and multiplicity of first responders, destinations, and obstacles;
- a multi-agent system, which supports information collection from hazard simulations, spatio-temporal data processing and analysis, connection with a geo-database, and route generation in dynamic environments affected by disasters;
- data models, which structure the information required for finding paths among moving obstacles, capturing both static information, such as the type of the response team, the topology of the road network, and dynamic information, such as changing availabilities of roads during disasters, the uncertainty of the moving obstacles generated from hazard simulations, and the position of the vehicle;
- path planning algorithms, which generate routes for one or more responders in the presence of moving obstacles. Using the speed of vehicles, departure time, and the predicted information about the state of the road network, etc., three versions (I, II, and III) of Moving Obstacle Avoiding A* (MOAAStar) algorithms are developed: 1). MOAAstar–I/Non-waiting, which supports path planning in the case of forest fires; 2). MOAAstar–II/Waiting, which introduces waiting options to avoid moving obstacles like plumes; 3). MOAAstar–III/Uncertainty, which can handle the uncertainty in predictions of moving obstacles and incorporate the profile of responders into the routing.

We have applied the developed prototype navigation system to different navigation cases with moving obstacles. The main conclusions drawn from our applications are summarized as follows:

- In the proposed taxonomy, we have identified 16 navigation cases that could occur in disaster response and need to be investigated. In addressing these navigation problems, it would be quite useful to employ computer simulations and models, which can make reliable predicted information about responders, the targets, and obstacles, in finding safe routes for the responders.

- The approach we provide is general and not limited to the cases of plumes and fires. In our data model, the data about the movement of hazards is represented as moving polygons. This allows the data model to be easily adjusted to merge and organize information from models of different types of disasters. For example, the areas that are affected by floods can also be represented as moving polygons. To facilitate the route calculation, not only the data of obstacles but also the information about the state of road networks affected by obstacles need to be structured and stored in the database.

- In planning routes for responders, the routing algorithms should incorporate the dynamic data of obstacles to be able to avoid the hazards. Besides, other factors, such as the operation time of tasks, the required arrival time, and departure time, also need to be considered to achieve the objectives in a rescue process, e.g., to minimize the delays caused by the moving obstacles.

- The profile of responders is quite important for generation of feasible routes for a specific disaster situation. The responders may have different protective equipment that allows them to pass through different types of moving obstacles, and thus can have different classification of risk levels to define the state of the road network. By taking into account the profile of the responders, the navigation system can propose customized and safe routes to them, which would facilitate their disaster response processes.

On the basis of our findings, we suggest the following topics for future work:

- As presented Wang and Zlatanova (2013c), there are still a couple of navigation cases that need to be addressed, especially the ones that involve dynamic destinations. More algorithms would be needed to solve these navigation problems. Besides, some extreme cases (e.g., the obstacle covers the target point during the course of an incident) also need to be investigated.

- Using standard Web services, an Android navigation application, which can provide navigation services in the environment affected by hazards, needs to be developed and tested in both the daily practice and real disasters. In this application, a user interface with various styling options should also be designed for different situations, e.g., waiting and moving, day and night, and urgent and non-urgent.
- Because the communication infrastructure may not be available or work properly during a disaster response, a decentralized method is needed to allow different users to negotiate with each other and to make local agreements on the distribution of tasks in case there is no support from the central planning system. Another type of multi-agent system would be needed to handle this situation.
- Introduce variable traveling speed into the re-routing process. The vehicle speed plays an important role in generation of routes avoiding moving obstacle, and can be influenced by many factors, such as the obstacles, the type of vehicles, traffic conditions, and the type of roads. Therefore, it would be needed to investigate how to derive the current and future speed from trajectories of vehicles.
- Apply the system to aid navigation in various types of natural disasters, using different hazard simulation models (e.g., flood model). More types of agents would be needed and integrated into the system to handle heterogeneous data from these models. Extensions of the data model are also required to meet a wider range of informational needs when multiple disasters occur simultaneously.

# SAMENVATTING

## Routeplanning voor eerstehulpdiensten rond bewegende obstakels

Navigatiediensten zijn steeds belangrijker geworden voor allerlei menselijke activiteiten, van het rondleiden van toeristen tot het ondersteunen van reddingsoperaties tijdens rampenbestrijding. Maar ondanks het feit dat er een aanzienlijke hoeveelheid onderzoek verricht is naar navigatie begeleiding, zijn er nog vele onderwerpen met betrekking tot de navigatie van eerstehulpdiensten die aandacht verdienen.

Tijdens rampen kunnen er verschillende typen bewegende obstakels ontstaan (zoals branden, rookwolken en overstromingen) die delen van het wegennet tijdelijk ontoegankelijk maken. Hulpdiensten moeten van plek naar plek navigeren om hun taak te kunnen volbrengen in een veranderende omgeving die geteisterd wordt door de ramp. Hiervoor hebben zij een routeplanner nodig die om kan gaan met dergelijke bewegende obstakels en die tegelijkertijd snel en efficiënt is.

Gedurende de laatste tientallen jaren zijn er steeds meer simulaties ontwikkeld die modellen sturen met dynamische data uit het veld. Deze simulaties gebruiken methodes zoals data assimilatie, stochastische schattingen en sturing van de data-inwinning door meettechnieken die zich aan de omstandigheden aanpassen. Deze simulaties kunnen betrouwbaardere resultaten van gevaren genereren en waardevolle informatie leveren over de staat van het door rampen geteisterde wegennet, ter ondersteuning van de routeplanning van eerstehulpdiensten die zich te midden van de bewegende obstakels bevinden.

Het doel van dit onderzoek is het ontwikkelen van een geïntegreerd navigatiesysteem voor eerstehulpdiensten die bewegende obstakels moeten vermijden. Een dergelijk systeem moet, gebruikmakend van uit simulaties afgeleide voorspellingen over de bewegende obstakels, in staat zijn de routeplanning te ondersteunen voor één of meerdere hulpverleners naar één of meerdere bestemmingen, terwijl bewegende obstakels vermeden worden. Dit proefschrift gaat uit van de volgende onderzoeksvraag:

*Hoe kunnen één of meerdere eerstehulpverleners veilig en efficiënt naar één of meerdere bestemmingen navigeren terwijl bewegende obstakels vermeden worden?*

Om de bovenstaande onderzoeksvraag te kunnen beantwoorden is dit onderzoek als volgt gestructureerd: 1). literatuuronderzoek; 2). conceptueel ontwerp en analyse; 3). implementatie van het prototype; en 4). beoordeling en aanpassing van het prototype. We hebben voorgaand onderzoek met betrekking tot navigatie bij rampen bestudeerd, en een geïntegreerde navigatiesysteemarchitectuur ontworpen dat hulpverleners ondersteunt in de verwerking, analyse en opslag van ruimtelijke informatie. Met behulp van deze architectuur zetten we simulaties in om voorspellingen te kunnen maken over obstakels en kiezen we een geo-database uit om de data op te slaan die nodig is voor de navigatie tijdens rampenbestrijding. Tijdens de ontwikkeling van het navigatiesysteem hebben we geïntroduceerd:

- een taxonomie van verschillende typen navigatiesituaties te midden van obstakels, waarin categorieën gemaakt zijn op basis van het type en het aantal hulpverleners, de bestemmingen en de obstakels;
- een multi-agent systeem met ondersteuning voor informatie-inzameling uit simulaties, ruimtelijk temporale dataverwerking en -analyse, verbinden met een geo-database en routeplanning in dynamische omgevingen getroffen door rampen.
- datamodellen, die de informatie die benodigd is voor routeplanning te midden van bewegende obstakels structureert, gebruikmakend van zowel statische informatie (zoals het type hulpdienst en de topologie van het wegennet) als dynamische informatie (zoals de wisselende beschikbaarheid van wegen tijdens een ramp), de onzekerheid in gesimuleerde bewegende obstakels en de posities van de voertuigen.
- routeplanningsalgoritmes, die routes generen voor een of meerdere hulpverleners rond bewegende obstakels. Gebruikmakend van onder andere de snelheid van voertuigen, vertrektijd, en voorspellingen over de toestand van het wegennet, etc., zijn er drie varianten (I, II, en III) van het Moving Obstacle Avoiding A* (MOAAStar) algoritme ontwikkelt. 1). MOAAstar–I/Non-waiting, voor routeplanning rond bewegende obstakels zonder wachtoptie voertuig(te gebruiken bij bijvoorbeeld bosbranden; 2). MOAAstar–II/Waiting, met wachtopties om bewegende obstakels zoals giftige rookwolken te vermijden; 3). MOAAstar–III/Uncertainty, dat kan omgaan met onzekerheid in de voorspelling van bewegende obstakels en rekening houdt met het type hulpverleners.

We hebben een prototype van een multi-agent navigatiesysteem geïmplementeerd en toegepast op vier navigatiesituaties met bewegende obstakels. De belangrijkste conclusies die hieruit volgden zijn als volgt:

- In de voorgestelde taxonomie zijn er zestien mogelijke navigatiesituaties die onderzocht moeten worden. In de aanpak van deze navigatieproblemen is het zeer nuttig om computersimulaties en -modellen in te zetten die betrouwbare voorspellingen kunnen maken voor hulpverleners, over bestemmingen en obstakels tijdens het vinden van veilige routes.

- De aanpak die wij voorstellen is breed inzetbaar en niet gelimiteerd tot situaties met rookwolken en branden. In ons model worden de data over de verplaatsing van gevaren gemodelleerd als bewegende polygonen. Dit maakt het datamodel makkelijk aanpasbaar en integreerbaar met informatie van modellen van andere typen rampen. Zo kunnen bijvoorbeeld de gebieden die getroffen zijn door overstromingen kunnen ook worden gemodelleerd als bewegende polygonen. Ter ondersteuning van de routeberekening moet niet alleen informatie over obstakels maar ook informatie over de toestand van het wegennet met obstakels worden gestructureerd en worden opgeslagen in een database.

- In de routeplanningsalgoritmes voor hulpverleners, moet rekening gehouden worden met dynamische informatie over obstakels om gevaren te kunnen vermijden. Bovendien moeten andere factoren zoals de benodigde tijd voor het uitvoeren van taken, de vereiste aankomsttijd en het tijdstip van vertrek, ook in overweging genomen worden om de vertraging door de bewegende obstakels te minimaliseren.

- Het profiel van een hulpverlener is belangrijk voor het genereren van geschikte routes voor een specifieke rampsituatie. Hulpverleners kunnen namelijk de beschikking hebben over beschermende uitrusting zijn die het mogelijk maakt zich door de bewegende obstakels heen te bewegen. Dit verandert de classificatie van risiconiveaus over de toestand van het wegennet. Door het profiel van hulpverleners in overweging te nemen, kan het navigatiesysteem een veilige route op maat voorstellen die hen helpt in de rampenbestrijding.

Op basis van onze bevindingen stellen wij de volgende onderwerpen voor toekomstig onderzoek voor:

- Zoals beschreven Wang and Zlatanova (2013c), zijn er nog een aantal navigatiesituaties die onderzocht moeten worden, met name de situaties met dynamische bestemmingen. Er zijn meer algoritmes nodig om deze navigatieproblemen op te lossen. Ook moeten uitzonderlijke situaties (zoals obstakels die tijdens een incident een bestemming afdekken) nader onderzocht worden.

- Met behulp van standaard webdiensten moet er een Androidapplicatie voor navigatiediensten in een door rampen getroffen omgeving worden ontwikkeld en worden getest in zowel de dagelijkse praktijk als tijdens echte rampen. Deze applicatie zou een user interface moeten hebben met kleuren en vormgeving die afhankelijk is van de verschillende situaties, bijvoorbeeld wachten en bewegen, dag en nacht, en urgent en niet urgent.

- Omdat de communicatie infrastructuur tijdens een ramp mogelijk slecht werkt of niet beschikbaar is, moet er een gedecentraliseerde methode komen die verschillende gebruikers met elkaar laat samenwerken in de verdeling van taken in het geval dat er geen ondersteuning van een centrale dienst is. Hiervoor zijn andere soorten multi-agent systemen nodig.

- Neem een variabele snelheid op in het herberekeningsproces van routes. De snelheid van een voertuig speelt een belangrijke rol in de generatie van routes die bewegende obstakels vermijden. Deze snelheid kan beïnvloed worden door vele factoren, zoals obstakels, het type voertuig, het verkeer, en het type en de toestand van de weg. Daarom zou onderzocht moeten worden hoe de snelheid van dat moment en de toekomstige snelheid van de route van een voertuig afgeleid kan worden.

- Het systeem toepassen om de navigatie te begeleiden in verschillende typen natuurlijke rampen, gebruikmakend van verschillende simulatiemodellen (bijvoorbeeld het overstromingsmodel). Meerdere soorten agents zouden nodig zijn en in het systeem geïntegreerd moeten worden om om te gaan met de heterogene data van deze modellen. Uitbreidingen van het datamodel zijn ook nodig om te kunnen voorzien in de grotere informatie behoefte in het geval meerdere rampen zich tegelijk voltrekken.

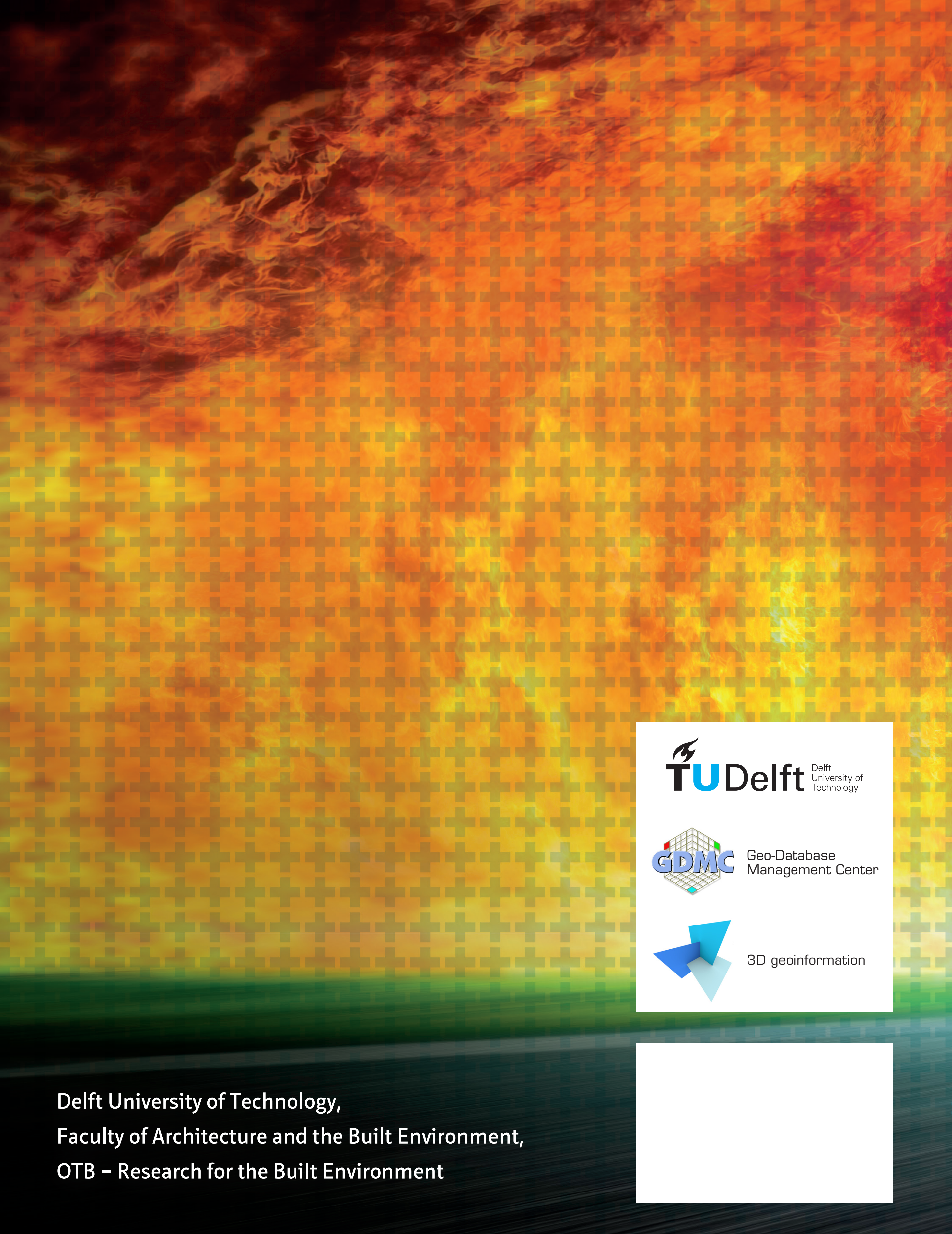
# CURRICULUM VITAE

Zhiyong Wang was born in Xingning, Guangdong, China, on January 02, 1984. From 2003 to 2007, he studied in Northeastern University, Shenyang, China, and received his Bachelor of Engineering degree in Automation. After that, in 2007 he continued his Maser study in Tongji University, Shanghai, China, and obtained his Master degree in Control Theory and Control Engineering. Funded by China Scholarship Council (CSC), in 2011 he started his PhD research topic “Navigation for first responders” in Delft University of Technology, Delft, The Netherlands, under the supervision of Prof. Peter van Oosterom and Dr. Sisi Zlatanova.

During his PhD study, he focused on the approaches supporting path planning for first responder in the presence of moving obstacles. In October 2012, he visited Vicomtech, San Sebastian, Spain. During the visiting, he investigated the use of fire simulation to support route planning in the case of forest fires. This work was supported by COST Action TU0801 “Semantic Enrichment of 3D City Models for Sustainable Urban Development". In March 2015, he started working on the research topic "Integrated indoor/outdoor navigation among moving obstacles", which was supported by Wuhan University-TU Delft Joint Research Centre on Spatial Information.

# LIST OF PUBLICATIONS

| No. | Publication (sequenced by publication date) | Related chapters |
| --- | --- | --- |
| 1. | Wang, Z. and Zlatanova, S. (2013c). Taxonomy of navigation for first responders. In *Progress in Location-Based Services*, pages 297–315. Springer | 3 |
| 2. | Wang, Z. and Zlatanova, S. (2013a). An A*-based search approach for navigation among moving obstacles. In *Intelligent Systems for Crisis Management*, pages 17–30. Springer | 6 |
| 3. | Wang, Z. and Zlatanova, S. (2013b). Multi-agent infrastructure assisting navigation for first responders. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science*, IWCTS '13, pages 1–6, New York, NY, USA. ACM | 5 |
| 4. | Wang, Z., Zlatanova, S., Moreno, A., van Oosterom, P., and Toro, C. (2014). A data model for route planning in the case of forest fires. *Computers* & *Geosciences*, 68:1–10 | 4, 6 |
| 5. | Wang, Z. and Zlatanova, S. (2014). Multi-agent based path planning for first responders among moving obstacles. *Computers, Environment and Urban Systems*. under review | 4, 5, 6 |
| 6. | Wang, Z., Zlatanova, S., and van Oosterom, P. (2015). Path planning for first responders among uncertain moving obstacles. Manuscript to be submitted to an international scientific journal | 4, 6 |



Delft University of Technology,
Faculty of Architecture and the Built Environment,
OTB – Research for the Built Environment