An Infrastructure Based Worm Spreading Countermeasure for Vehicular Ad Hoc Networks

by

Qi Zhang

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

VANETs are the essential component of the intelligent transport system, which attract research and industrial interests increasingly. As the multifunctional mobile nodes integrating transporting, sensing, information processing, and wireless communication capabilities, vehicular nodes are facing remarkable security issues and more vulnerable to malware attack than conventional communication nodes. In this thesis, the behavior and the security issues of the worm spreading on VANETs are studied. The approaches of the worm spreading on VANETs are discussed and an infrastructure based worm containment strategy is proposed. The infrastructure based worm containment problem is modeled as minimum contamination problem by introducing the expected contamination degree. Then the existing greedy method is applied to solve the proposed problem in VANETs scenario. After that, the Grid-shrinking Greedy Method and the Simplified Greedy Method are proposed which incorporate the characteristics of road networks and VANETs respectively. Simulation results show the two proposed methods outperform the existing greedy method and the comparison method from both complexity and solution quality aspects.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor Prof. Azzedine Boukerche for his support while I was in difficulties, guidance while I was confused, and encouragement while I was hesitating. His motivation and enthusiasm profoundly affected me and pushed me to overcome every encumbrance. More importantly, his generous financial support makes me concentrate on my research. I am honored to be a member of the PARADISE Laboratory.

Secondly, I would like to state my genuine thankfulness to Dr. Xin Fei for his persistent tutoring, straightforward but instrumental comments, and understanding in the past two years. It was a great luck to meet him in the PARADISE Laboratory.

Last but not the least, I would like to thank my families for their unconditional support, help, understandings, and love to make things happened.

The map data used in this research is downloaded from the OpenStreetMap. Map data copyrighted OpenStreetMap contributors and available from http://www.openstreetmap.org.

Table of Contents

List of Tables vii

List of Figures viii

Nomenclature ix

1 Introduction 1

1.1 Motivation 1

1.2 Objectives and Contributions 3

1.3 Outline 4

2 Literature Review 6

2.1 Malware and Computer Worms 6

2.1.1 Computer Worms 7

2.1.2 Classification of Computer Worms 8

2.1.3 Worm Spreading Model for the Internet 10

2.2 Worms on VANETs 13

2.2.1 Target Finding 14

2.2.2 Worm Transferring 16

2.3 Modeling Worm Spreading on VANETs 17

2.3.1 Vehicle Mobility Model 17

2.3.2 Analytical Method 22

2.3.3 Data-driven Method 26

2.3.4 Simulator Based Method . . . 30
2.4 Worm Containment Strategy . . . 31
2.4.1 Preemptive and Interactive Immunization . . . 32
2.4.2 Blacklist Isolation . . . 34
2.5 Conclusions and Open Issues . . . 34

3 Worm Containment on VANETs **36**
3.1 Worm Spreading on VANETs . . . 37
3.1.1 Approach of Worm Spreading . . . 37
3.1.2 Worm Propagation on Road Networks . . . 38
3.2 Worm Containment by Blacklist . . . 39
3.2.1 Worm Sniffing . . . 40
3.2.2 Blacklist Immunization . . . 40

4 MVSU deployment problem **42**
4.1 Worm Diffusion on Road Networks . . . 43
4.1.1 Worm Spreading by V2V Communication . . . 43
4.1.2 Vehicle Mobility . . . 44
4.1.3 Problem Formulation . . . 45
4.2 Contamination Model on Graphs . . . 46
4.2.1 Build Graphs from Map Data . . . 47
4.2.2 Minimum Contamination Problem . . . 48

5 Proposed Methods **52**
5.1 Original Greedy Method (OGM) . . . 52
5.1.1 Bond Percolation and Percolation Threshold . . . 53
5.1.2 Bond Percolation Approximation . . . 55
5.1.3 Algorithm Description of OGM . . . 57
5.2 Grid-shrinking Greedy Method (GGM) . . . 59
5.2.1 Grid Detection and Shrinking . . . 59

5.2.2 Edge Mapping 64
5.2.3 Algorithm Description of GGM 65
5.3 Simplified Greedy Method (SGM) 67
5.3.1 Equivalence of Directed and Undirected Graphs 68
5.3.2 Algorithm Description of SGM 69

6 Experimental Results **73**
6.1 Simulation Environment 73
6.2 Experimental Settings 75
6.3 Scalability and adaptability 80
6.4 Degree Distribution and Percolation Threshold 82
6.5 Contamination Degree 83
6.6 Average Blocking Probability 89
6.6.1 Solutions Found by MECD and MWCD 91
6.6.2 Shortest Step Path Blocking Probability 91

7 Conclusions and Future Work **97**
7.1 Conclusions 97
7.2 Future Work 98

References **100**

List of Tables

2.1 Summary of Worm Classifications 9

2.2 Worms on the Internet and VANTs 15

2.3 Summary of Discussed Models 20

2.4 Factors being Considered in Models 21

2.5 Countermeasures and Evaluation Matrix 32

2.6 Factors and Their Influence on the Speedy of Worm Spreading 35

6.1 Summary of Road Graphs Used for Simulation 75

6.2 Percolation Threshold and M for Each Method 78

6.3 Edge Cut Found by Max-flow Method 80

List of Figures

4.1 Original Map and the Graph after Simplification 49

5.1 Bond Percolation Threshold of the Grid 61

6.1 Graphs Built from Map Data 74
6.2 Grid Detection Results of Map 1 to 6 76
6.3 Grid Shrinking Results of Map 1 to 6 77
6.4 Bond Percolation Threshold (Directed Graphs) 79
6.5 Complexity Comparison 82
6.6 Degree Distribution before and after Grid Shrinking 84
6.7 Bond Percolation Threshold Curves Comparison 85
6.8 Result Evaluation by Contamination Degree 88
6.9 Solution Comparison (Solutions to MECD) 89
6.10 Solution Comparison (Solutions to MWCD) 90
6.11 Shortest Step Path Blocking Probability (Solutions from OGM) 92
6.12 Shortest Step Path Blocking Probability (Solutions from GGM) 92
6.13 Shortest Step Path Blocking Probability (Solutions from SGM) 93
6.14 Blocking Probability (MECD on Shortest Step Paths) 94
6.15 Solution Comparison by Blocking Probability 95

Nomenclature

Abbreviations

CC	Connected Component
GGM	Grid-shrinking Greedy Method
I2V	Infrastructure-to-Vehicle
MCP	Minimum Contamination Problem
MECD	Minimum Expected Contamination Degree
MVBL	Malicious Vehicle Blacklist
MVSU	Malicious Vehicle Screening Unit
MWCD	Minimum Worst Contamination Degree
OGM	Original Greedy Method
RSU	Road Side Unit
SGM	Simplified Greedy Method
SLCC	Second Largest Connected Component
SLSCC	Second Largest Strongly Connected Component
SCC	Strongly Connected Component
SSP	Shortest Step Path
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad Hoc Network

Mathematical Symbols

C	contamination degree
$\overrightarrow{e}_i$	arbitrary directed edge
e_i	arbitrary undirected edge
$\overrightarrow{e}_{uv}$	directed edge from node u to node v
e_{uv}	undirected edge from node u to node v

$\overrightarrow{G_p}$	directed graph generated by bond percolation process
G_p	undirected graph generated by bond percolation process
$N(v_i)$	the neighbor nodes of node v_i
p_p	bond percolation probability
p_{th}	bond percolation threshold
$p(v_i)$	probability distribution of node v_i

Chapter 1

Introduction

Intelligent transport and automated driving are so attractive that the research on VANETs and V2V telecommunication are blooming in recent years. The V2V communication enables the vehicles on the road to share information with surrounding vehicles and to access the Internet. With the help of sensing and wireless communication, the driver assistance system and the automated driving system could come to true. The V2V communication will also significantly reduce the risk of traffic accidents. The drivers can make decisions based on the shared information and respond to a road incidence or a traffic accident before visual contact. The VANETs enabled vehicles will provide the functions such as driver assistance, automated driving, onboard multimedia, and onboard Internet applications [54], which would remarkably enrich the passages' onboard experience.

1.1 Motivation

The VANETs are ad hoc networks composed of the vehicular nodes, which is revolutionary to driving and transporting. However, the ad hoc vehicular networks also bring cyber security and road safety issues on driving. Some cases have been reported that onboard computers can be overridden by third-party via the wireless connection [61]. The override control during driving could cause a seriously public safety issue. Unauthorized access to a vehicular node could lead to privacy and security issues. Many studies have been carried out to address the security issues of VANETs from diverse aspects [81] [24] [23]. However, in contrast to the flourish of the security protocols and frameworks studies [19] [33] [41] [32], the worm spreading on VANETs is less studied.

Worm threat is originated from computer networks but has been found in almost every existing data-enabled networks, such as the Internet, Bluetooth networks, and cellular

networks [27]. It is predictable that the emergence of worm threat in VANETs is inevitable.

A computer worm is a type of malware that spreads rapidly among the networks [60]. A computer worm is capable of self-production and self-propagation without human intervention, so the worm infection is hard to be aware by users. The behavior of worm follows a predefined rule which repeats on each newly infected node. This feature can be utilized to detect and predict the worm propagation. To inhibit the worm spreading, it is essential to study the worm propagation pattern. Many studies of modeling the worm spreading on the computer networks and mobile networks have been carried out and a number of containment techniques are proposed [63] [98] [59].

Along with the development of wireless communication, the worms spread not only on the conventional computer networks but also on the mobile networks such as Wi-Fi access points and data enabled cellular networks. Gu et al. [62] studied the worm spreading on the metropolitan wireless networks. Their research shows that the overlapped wireless access points could be a potential path of worm propagation. Sarat and Terzis [85] investigated the influence of user mobility on the malware propagation. They concluded that the malware spread rapidly among mobile nodes, and the mobile nodes could impact hundreds of network domains in a short period of time. Similarly, Yanmaz [96] proposed a study on the impact of mobility on the worm spreading in static networks. Zhai et al. [98] focused on the networks being composed of mobile phones and computer hosts. They proposed an analytical study for modeling the worm propagation on heterogeneous networks. Adu-Gyamfi et al. [4] proposed the model for the spreading of passive worms in mobile social networks. Bulygin studied [43] the worms spreading by Bluetooth and MMS (Multimedia Messaging Service) message. Tang et al. [86] studied the mobile malware propagation and proposed a time-aware scheme containment strategy.

However, the worm propagation behavior is unique in VANETs due to the vehicle mobility and V2V communication. As a result, the worm spreading models and research conclusions for conventional networks may not hold in VANETs scenarios [88]. In recent years, some studies on the worm spreading on VANETs have been carried out. Most of them focus on analyzing and modeling the propagation process but the countermeasures. Although some immunization patching based countermeasures have been proposed, the infrastructure deployment strategy is hardly discussed. Our research is motivated by the threat of worm spreading on VANETs, and we address both the worm propagation behavior and the special infrastructures deployment strategy for inhibiting the worm outbreak on VANETs.

1.2 Objectives and Contributions

The primary objectives of our study are to investigate the worm propagation on VANETs and to suggest an approach inhibiting the worm spreading on VANETs. The proposed method should have an excellent scalability to apply on large-scale networks. To achieve this goal, the analysis of worm propagation behaviors in VANET environment are required. The existing approaches for modeling worm spreading on VANETs are discussed first. Then the Malicious Vehicle Screening Unit (MVSU) deployment problem is proposed to formulate our optimization objective. After that, the proposed MVSU deployment problem is modeled as minimum contamination problem. We first transplant the existing greedy method and apply it to VANETs environment. Then, two greedy methods are proposed to solve the minimum contamination problem on a road network and find a solution to MVSU deployment problem.

Our contributions are summarized as follows:

- A classification of existing worm spreading models of VANETs are suggested. To the best of my knowledge, this study is the first systematic review of the existing studies for worm propagation in VANETs' environment. The existing models are categorized based on their study methods. The parameters, features, and limitations of each model are summarized.

- A macroscopic analysis method is suggested for worm spreading on VANETs. The worm spreading by V2V communication and physical carrying-spreading are discussed respectively. From a macroscopic point of view, both of the two spreading approaches propagate worm on a network constrained by the underlying road network. By the proposed macroscopic method, the worm spreading processes of V2V communication and physical carrying-spreading are combined and analyzed as one information diffusion process on road networks.

- Information diffusion model is introduced to describe vehicle mobility. The vehicle mobility is modeled as an information propagation process on a road network. The prime approach is employed to build a directed graph based on the road topology. The initial location and potential destinations are modeled as the source node and pass-through probability of each edge. Then the vehicle mobility is modeled as a diffusion process on a directed graph which facilitates the study of worm propagation on VANETs.

- The contamination control model is extended by introducing the expected contamination minimization problem. The contamination minimization problem proposed

by Kimura et al. [70] is for the scenarios that the location of contamination source node follows the uniform distribution. However, the location of the source node is not necessarily following the uniform distribution in the real world. The expected contamination degree is defined to overcome this limitation. It incorporates the probability distribution of the initial location of contamination source node, so the applicable scenarios of contamination control model are extended.

- Propose a Grid-shrinking Greedy Method which is an enhanced greedy method for some particular types of road networks. The existing greedy method was proposed for solving the minimum contamination problem on social networks and computer networks [70]. We study the minimum contamination problem on road networks which possess unique characteristics. The proposed Grid-shrinking Greedy Method takes advantage of the common road pattern and outperforms the existing greedy method for some types of road networks regarding the better solution quality and reduced complexity.

- A Simplified Greedy Method is developed to solve the minimum contamination problem road networks. The proposed simplified greedy method gets over the limitation of Grid-shrinking Greedy Method and outperforms both the Grid-shrinking Greedy Method and the Original Greedy Method by finding a better solution with lower complexity.

- Extensive experiments on different road networks are implemented to verify the proposed worm spreading analysis method. The simulation is based on real map data with diversity characteristics. The simulation results show the robustness of the proposed methods and the influence of road topologies on the worm spreading containment.

1.3 Outline

The outline of this thesis is summarized as follows. A literature review is given in Chapter 2 where the possible approaches of worm epidemic spreading on VANETs is investigated, and the methods used for worm spreading studies are discussed. In Chapter 3, the worm spreading behaviors in VANETs are analyzed first, and then an infrastructure based worm spreading countermeasure is proposed. In Chapter 4, the Malicious Vehicle Screening Unit (MVSU) deployment problem and the modeling procedure are introduced. The methods solving the MVSU deployment problem are proposed in Chapter 5. The performance of

the proposed methods is evaluated by simulations. The experimental settings and results are discussed in Chapter 6. The conclusions and future work are summarized in Chapter 7.

Chapter 2

Literature Review

The related work of worm spreading on the Internet and VANETs is reviewed in this chapter. Characteristics and classifications of computer worms are introduced in Section 2.1. The possible approaches of the worm spreading in VANETs are discussed in Section 2.2. A comprehensive review of the existing worm spreading models for VANETs' environment is presented in Section 2.3. After that, containment strategies for inhibiting worm propagation are introduced in Section 2.4. The conclusions and open issues are summarized in Section 2.5.

2.1 Malware and Computer Worms

Malware is a remarkable security issue in computer networks. The spreading of malware on the Internet has been intensively studied. Generally speaking, malware has three main types: virus, Trojans, and worms. Each kind of malware possesses unique characteristics. A virus usually attaches itself to a host program or a host file and will be active when users execute the host program or the infected file [27]. The virus usually does not interfere the normal function of the host program. The way of viruses spreading is passive since they can neither self-reproduce nor self-propagation. The virus propagates along with the copying or downloading of host program or infected file. Trojans is a kind of malware that is pretended as a normal software. The users are usually enticed to download and to execute the Trojan, so user intervention is required for Trojan spreading. Worms are a type of stand-alone malware that could be executed and propagated by itself [58]. Compared with virus and Trojans, independence is the unique characteristic of worms which enables the worm to propagate more rapidly than virus and Trojans. Although the propagation approaches of viruses, Trojans and worms are diverse, they cause some common security

threats, such as increasing network traffic, backdoor and file modification.

2.1.1 Computer Worms

Computer worms have become an ongoing threat to computer networks since its first appearance in 1988 [60]. Compared with viruses and Trojans, Worm epidemic is one of the most critical security threats in computer networks because of its self-copying propagation and concealment. Because the worm propagation is independent of users' intervention, the worms could outbreak in a short period of time.

The spreading of computer worms can impact not only the hosts on the Internet but also cause physical damage to manufacturers and industrial plants. Some of the worm attacks have resulted in disastrous consequences. For example, one of the deadliest computer worms named ILOVEYOU was created in the year 2000. This worm was propagated by email. Once it is triggered, the worm would email itself to the first 50 contacts in the address book of the system. This very worm infected a tremendous number of the computer hosts and caused a total damage of about $ 10 billion [77]. In 2001, the Code Red worm being designed for attacking web servers infected more than 359,000 hosts within one day. Till 2003, the accumulative loss caused by Code Red is about $ 2.6 billion [92]. In 2003, The Blaster worm targeting at hosts running Windows XP and Windows 2000 systems infected around 423,000 machines within four days [80]. Another worm named Mydoom appeared in 2004 which spread rapidly by mass-mailing itself to the email addresses collected from the host system. The hosts infected by Mydoom would launch denial of service attack. Because of the combination of the fast-spreading mass mailer worm and massive attack, the estimated damage caused by Mydoom reached up to $38 billion [82]. The worms could even become a cyber-weapon to paralyze the industrial plants. The Stuxnet is a highly sophisticated computer worm targeting programmable logic controllers (PLCs). The PLCs are widely used for automation control in factory assembly lines so that the worm can destroy the high-value industrial target [67]. The Stuxnet is believed as the first cyber weapon used to sabotage Irans nuclear program [84].

The life cycle of a typical computer worm can be defined as four phases: target finding, worm transferring, worm activation, and infection [75]. In the target discovery phase, a worm searches the reachable hosts in the network and selects one or more hosts as the target to be infected. In worm transferring phase, a worm sends a copy of worm body to the target hosts by a worm carrier. Once worm goes to infection phase, which means worm has successfully copied itself into the new host and been hibernating for the activation. A worm is triggered to be active by activation phase. After that, the worm starts a new target finding and propagation cycle. Worm infection and activation phases are local

behaviors having little interaction with other network components. The detection of worm activation and worm infection highly depends on the carrier and the worm characteristics. We focus on the worm spreading behavior which is determined by the target finding and worm transferring phases.

2.1.2 Classification of Computer Worms

Many classifications have been suggested according to different characteristics of computer worms. The worm classifications are discussed here to investigate the worm characteristics. Weaver et al. [94] presented an overview of worm classifications which cover every phase of worm life cycle. In [94], besides the worm characteristics, such as target finding schemes, worm carrier mechanisms, the approaches of activating, and effective payloads, the worm were also classified based on the motivation of creating the worms. The motives of creating the worms are not related to worm behaviors, so they are not in the scope of our study. The worm characteristic based classifications in [94] are influential to later studies and wieldy used. So we summarize some of the classifications in [94] and present them in Table 2.1.

Weaver et al. [94] classified the worms by target discovery strategies which include scanning, pre-generated target list, internal target list, passive, and application-dependent propagation. The scanning worms scan the target IP address space with or without preference. Some worms are designed for attacking a set of particular hosts, so a target list is employed. The target list is also called hit-list. The hit-list would be either embedded within the worm body or stored externally. For example, the hit-list could be stored in a server or a host which is accessible by the worms. Some worms generate an internal target list according to the local information on the host. Some worms do not actively seek for vulnerable hosts but wait for vulnerable hosts visiting the infectious host. This sort of worms is classified as passive worms. There are some worms propagated along with other application software. When the application software is popular, the worm spread itself along with the distribution of host application software.

Based on the carrier of worm propagation, Weaver et al. [94] categorized the worms into self-carried propagation, second channel propagation, and embedded propagation. The worms employing self-carried have a complete copy of the worm body. The worm body is propagated to vulnerable host straight forward from the infectious host. Some worms using the second channel to propagate the worm body. When the second channel is employed, the worm body would be stored at a server or some specified hosts. The infectious host does not propagate the entire worm body but just start a download process on the vulnerable hosts. The worm body is propagated when susceptible hosts download the worm body via the second channel. The worms spreading with files or other application software are

Table 2.1: Summary of Worm Classifications

Worm Characteristic	**Classification**	**Summary**
Target Discovery	Scanning	Worms probe a set of IP addresses to identify vulnerable hosts.
	Pre-generated Target List	The hit-list is embeded into the worm body before release.
	Externally Generated Target List	The hit-list is separated with the worm body but accessible by worms.
	Internal Target List	Worms build a hit-list based available information on local host.
	Passive	Worms wait for the visiting of vulnerable hosts. Users intervention reuqired.
	Application-dependent Worm	Worm is embedded in application. Hosts being installed the application will be infected.
Propagation Carrier	Self-Carried	Worms propagate the whole worm body.
	Second Channel	Worms trigger downloading, worm body is download from third party host.
	Embedded	The Worm body is embeded in host files or programs.
Approach of Activation	Human Activation	Users execute the worm itself to active the worm life cycle.
	Human Activity-Based Activation	Users execute other files or programs to trigger the worm life cycle.
	Scheduled Process Activation	Worms are activated by scheduled system processes.
	Self Activation	Worms are self-activated after infection.
Effective Payload	None/nonfunctional	Worms do nothing on the host, but waste system and network resources.
	Internet Remote Control	Worms open a backdoor on the host for unauthorized access and control.
	Spam-Relays	Worms create open-mail relay for spammers.
	HTML-Proxies	Worms distribute web-proxies
	Internet DOS	Worms can launch the denial of Service (DOS) attack.
	Data Collection	Worms collecte and disclose data on the host.
	Access for Sale	Worms allow remote asscess and online payment.
	Data Damage	Worms encrypt or destroy data on the host.
	Physical-world Remote Control	The worms cause influence in physical-world.
	Worm Maintenance	A worm is used to update existing worms.

categorized as embedded worms. The worm attached files are the propagation carriers of embedded worms.

According to the different activation methods, computer worms are classified into human activation, human activity based activation, scheduled process activation, and self-activation [94]. The worms activated by human intervention will not be infectious until the users execute the worm body. For Some worms, the activation is dependent on other applications. This kind of worms is activated when users run the applications being able to trigger the worm. The self-activation is commonly used as activation method of worms. Because there is no user intervention involved for the self-activation worms, the worm could propagate rapidly.

The worms are designed for implementing some purposes. The code to fulfill a certain task is called effective payload. Weaver et al. [94] categorized the worms based on their effective payloads, such as nonfunctional payload, the internet remote control, spam-relay, HTML-proxies, Internet DOS, data collection, data damage, physical-world remote control, and worm maintenance. Note that the nonfunctional payload does not mean the worm is unharmful. The worms propagating and running among the susceptible hosts take the network throughput and system resources especially when a significant number of hosts are infected in a particular network. A brief of each category mentioned in [94] can be found in Table 2.1. Note that the size of the effective payload is influential to worm propagation. According to the simulation results by Abdelsamee [1], a smaller payload increases the speed of worm propagation. This conclusion agrees with the fact that a considerable number of worms employ nonfunctional payload to achieve a rapid outbreak.

The approaches of finding victim hosts are commonly used for worm classification. For example, with the respect that most hit-list based worms are also capable of IP scanning, Wang et al. [92] categorized target finding approaches by scan-based schemes and topology-based schemes. The scan-based schemes work well without topology information, which has two subtypes: the random scanning and localized scanning. Each subtype of the scan based schemes is detailed and further discussed in [92]. Zou et al. [102] covered the scan-based worm prorogation models by the different scanning schemes which included idealized worms, uniform scan worms, local preference scan worms, sequential scan worms, and selective attack worms.

2.1.3 Worm Spreading Model for the Internet

The worm propagation on the Internet has been intensively studied and many propagation models have been proposed. Instead of comprehensively introducing the Internet worm

propagation models, a brief of the modeling for worm spreading on the Internet is presented here.

The idealized worm models proposed by Zou et al. [92] describe the perfect scenarios of worm spreading, which include perfect worm model and flash worm model. A perfect worm has the complete IP address of all vulnerable hosts. During their propagation, the infectious hosts can coordinate their scanning process. It means the host already infected will not be scanned by other infectious hosts. The flash worm also has a full list of all vulnerable hosts. Instead of perfect coordinating, the flash worm is capable to uniformly select a target from the IP address list containing complete susceptible hosts. For both idealized worm models, the delay between two successive scanning is considered as the key factor impacting on the speed of perfect worm propagation. So the idealized worm models are characterized by the average scan rate and the scanning delay. A numerical analysis in [102] shows that for a given average scan rate, the propagation speed of perfect worm is slightly faster than the flash worm. Because the uniform scanning scheme used in flash worm cannot avoid scanning the hosts that already being infected, there are wasted scan during the scanning process. Although a worm satisfies the idealized worm models does not exist in the real world, the idealized worm models provide a general picture of how the different conditions would impact on the worm propagation.

According to the discussion by Wang et al. [92], the uniform scanning, hit-list scanning and routable scanning could be used by random scanning scheme. The sample worm of each random scanning scheme can be found in the real world. The uniform scanning is commonly used when the network topology and components information is unknown. The worms employing uniform scanning, such as Code Rid, randomly select an IP address from the whole IPv4 address space to be the target host. Ideally, every IP address in the address space has equal probability to be selected, but it is channeling to achieve the perfect randomness during the address selection. Besides, the large IP address space also limits the propagation speed of worm using uniform scanning scheme.

The uniform scan worm model by Wang et al. [92] characterizes the uniform scanning worms. The uniform scan worm model is distinguished from the uniform scan in the flash model because the IP address list in uniform scan worm model does not have the complete IP addresses of all vulnerable hosts. The IP address list used in uniform scan worm model is just a list of candidate targets which may contain both susceptible and immune hosts. For example, the Code Red worm select a target host from the entire IPv4 address space, regardless it is susceptible or immune.

The hit-list scanning and routable scanning were proposed to overcome the limitations of uniform scanning. These two schemes were also discussed in [92]. Hit-list scanning is an

effective random scanning scheme with reduced candidate IP address space. This scheme combines the uniform scanning with an IP address list. The worm first attacks the hosts in the hit-list, then it starts the uniform scanning to find other susceptible hosts. Because the hosts on hit-list would be infected rapidly, the hosts with high value or the hosts with a large number of reachable nodes could be listed in the hit-list to accelerate the worm spreading. Routable scanning scheme scans the IP addresses in the routable addresses among the IP address space. For example, the Border Gateway Protocol (BGP) routing prefixes can be used to find the potentially reachable nodes. By routable scanning, the number of IP address classes is significant reduced. But due to the extra size of BGP prefixes, the packet number required for successfully infecting a vulnerable host is increased. Another type of special hit-list scanning worms is named divide-and-conquer scan worms which split the IP address space into some narrower ranges and assign them to different infectious hosts to implement the scanning individually.

Wang et al. [92] categorized the localized scanning scheme into local preference scanning, local preference sequential scanning, and selective scanning. The local preference scanning takes the advantage that the IP addresses are a non-uniform distribution in the real world. The worm using local preference scanning scheme could either scan the different IP address space with different probabilities or scan the different IP address Classes with different probabilities. No matter which scheme is used, the objective is to scan the dense IP address range to find more susceptible hosts. The analytical study in [102] shows that the distribution of vulnerable hosts determines the effectiveness of local preference scan schemes. When the susceptible hosts are uniformly distributed in the scan space, the mathematical expression of the worm spreading speed of local preference scanning is the same as the uniform scanning. In this case, the speed of worm propagation by local preference scan and uniform scan are competitive. When the susceptible hosts follow the nonuniform distribution, the simulation result in [102] demonstrates the worm propagation speed of local preference scanning is faster than uniform scanning.

The worms with local preference sequential scanning scheme select a starting IP address with preference and then sequential scan the IP address within a given range. An analytical performance study of sequential scan worms is also carried out by Zou et al. [102]. In the sequential scan model, the scheme of selecting the starting IP address is influential to the speed of worm spreading. The speed of the worm spreading decreases when local preference scan is applied for finding the starting IP address. When the vulnerable hosts are uniformly distributed, the speed of worm spreading by sequential scan scheme is the same as the uniform scan scheme. When the susceptible hosts are not the uniform distribution, the rate of worm spreading is dependent on the individual distribution of vulnerable hosts.

The selective scanning focuses on the IP addresses within a particular address space.

The worms adopting selectively scanning scheme can target at a set of specific hosts based on the domain of their IP address. By this scanning scheme, it is possible to infect the hosts of a particular institution or certain area.

According to the discussion by Zou et al. [102], selective attack worms could be target-only or target-global. The target-only means the worms only scan the target domain and the newly infected hosts also only scope at the target domain. However, in the target-global scheme, the IP address space to be scanned by an infectious host is dependent on the domain where the infectious located. More specifically, the newly infected hosts within the target domain only scan the hosts in the target domain. The newly infected hosts in other domain perform a uniform scan of the global IP address space. The analytic study in [102] shows the target-global style selective attack worms are more effective because the infectious hosts outside the target domain have a chance to infect the hosts in the target domain by uniform scanning.

Note that although most existing models were developed based on IPv4 networks, the worms also threaten IPv6 networks. Zheng et al. [101] proposed a study on the comparison of the worm spreading on dual-stack networks and pure IPv4 networks. A dual-stack worm that is capable of spreading on both IPv4 and IPv6 is used in their study. Their simulation results illustrate that the dual-stack networks accelerate the worm spreading. Lin et al. [76] discussed the difference of worm spreadings on pure IPv4 and pure IPv6 networks. In their study, two worms designed for on IPv4 and IPv6 networks respectively are used for simulation. They concluded that the propagation speed of worms designed for IPv6 was faster than the one designed for IPv4 with comparable network parameters. Zulkiflee et al. [103] studied the behavior of an IPv4 worm spreading on the IPv6 network. Their simulation results show that the malware targeting at IPv4 networks can also propagate on IPv6 networks.

2.2 Worms on VANETs

Since the prosperous development of short-range wireless communication and mobile devices, the worm epidemic makers find their ways to gain benefit by spreading worms in mobile networks. It had been shown that the worms could be spread by point-to-point communication, such as Cabir worm spreading by Bluetooth [73]. With the widely use of mobile Internet and smart devices, the malware appears on every mainstream mobile operating system. Based on the history of worms propagation in existing mobile networks, it is believed that VANETs and vehicular nodes would be vulnerable to the worm attack. When this security threat appears in VANETs, it could cause severe road safety issues. Be-

cause the throughput of VANETs is limited, when the communication resource is occupied by malicious packets, the critical safety-related applications may be interfered. The back-door opened by worms on vehicular nodes is not only a security issue of personal privacy but also a threat to traffic and public safety. To better understand the worm spreading in VANETs, a comparison between the possible worm behaviors in VANETs and conventional computer networks is carried out.

2.2.1 Target Finding

The target finding phase of worms in VANETs should be different from conventional networks because of the dynamic topology and ad hoc characteristics [11]. In conventional computer networks, the blind scanning, predefined hit list, and topology awareness are the three dominant approaches to find vulnerable targets. Although the blind scanning may hit a wrong IP address, it is reliable and easy to implement [3]. Many technics can be employed to improve blind scanning scheme. For example, border gateway protocol (BGP) prefix can be used to narrow down the scanning range. The preference scanning takes advantage of the nonuniform distributed IP addresses in the real world to increase the probability of finding susceptible nodes [38]. The hit-list is another primary target finding scheme. The worm attempts to attack the targets on the hit list with higher priority.

In the conventional network structure, host list and routing table on network components also could be used for worm target finding. A worm could be aware the network topology by the information in the host list and routing table [19]. For example, the topology information can be utilized by topological worms to generate an infection path. The topology aware worms pretend the attack as sending a regular traffic along the routing path, so the attack is harder to be detected [15]. Besides, the attack launched by topology awareness worms could be more accurate and rapidly, because they could infect particular nodes or the nodes along a specific routing path.

The possible target finding schemes in VANETs are different from the conventional computer networks due to the characteristics of VANETs [34]. In the conventional computer networks, the target finding scheme relies on the IP routing. There is not physical distance limitation as long as the susceptible node is reachable by the infectious node. In VANETs environment, communication between vehicular nodes based on the multi-hop communication, the one hop infection distance is limited by the effective range of the V2V wireless communication [24]. Furthermore, the dynamic topology impedes the target finding schemes such as blind scanning and hit-list. The packets sent from an infectious node may never be able to reach a particular destination node due to the highly dynamic network topology of VANETs. For the same reason, the reachable nodes of an infected

Table 2.2: Worms on the Internet and VANTs

Worm Characteristics	**Internet Worm**	**VANET Worm**
Target Finding Scheme	IP scanning Hit List	Broadcasting
Propagation Scheme	Self-carried worms Second channel Embedded	Self-carried worms Second channel
One hop attack distance	Unlimited	<300m (typically)
Node Reachability (Topology)	Static	Dynamic
Transmission Delay	Low	High
Security threat	Increase network traffic Backdoor Modify files	Increase network traffic Backdoor Modify files Interfere critical applications

vehicular node are uncertain, so hit-list based and topology aware worms hardly work on VANETs [31]. In fact, most of the sophisticated target finding schemes cannot work in VANETs environment because of the limited contact time and available channel capacity between two adjacent vehicular nodes.

However, the multi-hop communication provides other approaches for worm spreading [29]. Broadcasting and unicasting are commonly used for V2V communication in VANETs. Because worm epidemic always greedily infects susceptible nodes, broadcasting is more suitable for worm spreading than unicasting [35]. In VANETs environment, an infectious vehicular node can find potential targets by broadcasting or sniffing the beacons sent by surrounding vehicles. With this kind of target finding schemes, all the nodes within effective communication range are susceptible [18]. Compared with the target finding schemes in conventional computer networks, worm attack in VANETs cannot be accurate since the potentially reachable nodes in ad hoc networks are varying along with time. The relatively short communication range and mobility constrain the one hop infection range. In other words, a worm epidemic spread by V2V multi-hop communication could be geographical locally restricted at initial stage rather than a global spreading. However, the broadcast based target finding scheme could cause an avalanche effect [14]. Each newly infected node could broadcast the infectious packets. As a result, the infected nodes number would increase exponentially.

2.2.2 Worm Transferring

After the susceptible host is identified, the worm starts to transfer itself to the target host. In conventional computer networks, the worm transferring approaches could be categorized into self-carried, second channel and embedded [94]. The self-carried worms send packets containing the worm body to the target hosts straight forward. Once the worm body is successfully propagated by self-carrying, the newly infected node has the potential to infected other nodes immediately. The spreading speed of a self-carried worm depends on the size of the worm and overall available bandwidth [54]. Instead of transferring the whole worm body, worms propagating by the second channel only trigger a download process on target hosts. The worm body is downloaded from another infected node or a server on the Internet. Note that the downloading activity is not necessarily started instantly after the infection [57]. The starting of the download process could be timer dependent or system environment dependent. For example, the download process could be started when a certain process is found in the infected host. The worm transferred by the second channel cannot reproduce itself until the worm body is fully downloaded, so there could be a latency between the infection and the worm outbreak [28]. The outbreak latency depends on the Internet accessibility and the trigger conditions. The embedded worm hides itself within a normal communication traffic, such as a message or a software installation package. To better cover itself, most embedded worms do not actively scanning susceptible targets but waiting for being downloaded by the potential targets. As a result, the spreading of embedded worms is highly dependent on users interactions.

The transmission delay is introduced to characterize the influence of the size of worm body to the worm spreading [100]. Transmission delay is defined by the time required for self-copying which depends on the size of worm body and the throughput of communication networks. Compared with the conventional computer network, V2V communication is latency tolerant and bandwidth limited, so the transmission delay in VANETs should be higher than the Internet in general [56]. Therefore, worms should take any opportunity to self-copying and propagating. The self-carried propagation is the most suitable scheme because the worm integrity is ensured. A newly infected node could be infectious immediately when the packets containing worm body are received [13]. The second channel scheme also works on VANETs, but the worm may be detected before it is activated due to the delay between infection and being infectious. The worm using embedded scheme has to be propagated together with host program or host file, which means more payload need to be transferred [22]. When the network throughput is limited, such as VANETs environment, more communication time is required. Therefore, embedded approach is unlikely to be applied to VANETs environment.

2.3 Modeling Worm Spreading on VANETs

The worm spreading model developed for conventional computer networks cannot be applied to dynamic networks such as VANETs [36]. Because the worm behaviors in VANETs are different from the worm behaviors in typical computer networks as discussed in Section 2.2, the existing worm spreading models for computer networks cannot represent the worm propagation in VANETs. The reasons can be summarized as followings: First, the network topology is dynamic, which is dependent of time and the traffic conditions [37]. Secondly, vehicle mobility determines the connectivity between vehicular nodes which means the worm can be prorogated by physical carrying-spreading. Thirdly, the worm propagation behavior would be different due to the unconventional target finding scheme and characteristics of ad hoc networks.

To study the worm spreading on VANETs, describing the underlying communication network is preliminary. The topology of VANETs can be defined by the combination of a vehicle mobility model and a communication channel model [81]. Vehicle mobility models represent the movement of vehicular, such as relative position and the varying distance between neighboring vehicular nodes. The vehicle mobility is influenced by the traffic volume, destinations, and road topology. Channel models describe the wireless communication reachability between two vehicular nodes [26]. The radio communication is governed by multiple factors, such as operating frequency, inter-reference, physical environment and transmission power, etc.. When the VANET topology is well defined, the worm spreading process could be characterized by applying a epidemic model to the topology.

In this section, we first introduce the classification of mobility models, and then the existing studies of modeling worm spreading are categorized based on their methodologies. The models, parameters, and conclusions of each method are introduced, and comments are given accordingly.

2.3.1 Vehicle Mobility Model

As a cornerstone of the VANETs research, vehicle mobility has been intensively studied [9] [14]. It governs how the dynamic topology of a VANET changes along with time, so the mobility factors are more influential on the worm spreading process rather than communication factors from a large time span aspect [40]. Therefore, we category the existing worm propagation studies for VANETs based on their mobility models. The mobility models can be categorized by their scopes, modeling methods, and model parameters.

According to the scope of the traffic mobility, vehicle mobility models can be classified into microscopic and macroscopic models. A microscopic model describes the movement

of each vehicular node, so its parameters are usually related to the individual node, such as velocity and position [21]. In contrast, the movement of a single vehicle is neglected in a macroscopic model. The traffic flow is described in a macroscopic point of view and characterized by some statistic parameters, such as vehicle density, vehicle distribution, average velocity, and average interval time or average distance, etc..

Based on the modeling methods, mobility models can be categorized to analytical models and data-driven models. In an analytically model, the statistic data from real traffic flow is unnecessary [30]. All the parameters can be determined by scenario assumptions. For example, vehicle positions are assumed to follow a certain probability distribution model, such as the uniform distribution, normal distribution, and the exponential distribution. Then, for a given road section, the average interval distance between adjacent vehicles can be estimated by the predefined vehicle distribution. However, the assumptions used in a analytical model do not hold all the time because of the diversity of VANETs scenarios. Even for the same scenario, the traffic conditions vary along with the time [17]. For example, the traffic conditions of rush hour and midnight in an urban scenario are quite different. To overcome the limitations of analytical models, some data-driven models are proposed. The historical statistic data or traffic monitor data from transport authority could be used to feed the data-driven models [10]. The data could be used either for the model calibration or for calculating some coefficients used in the model.

Note that the message transmission between two distant vehicles could rely on either purely multi-hop communication or physical carrying the message until they are near enough to fulfill multi-hop communication [41]. In the former case, the message transmission can be finished almost instantly, so the topology varying during message transmission is negligible. However, in the later case, the time for message transmission is long enough such that the dynamic topology should be taken into account [16]. In other words, the dynamic topology contributes to the success of message transmission. So the mobility models could be categorized by whether or not the change of network topology along with time is considered. A mobility model without time parameter is a static model which utilizes a snapshot of current vehicle distribution for topology analysis [39]. The model containing time parameter is a dynamic model that topology is varying along with time elapsing. A dynamic model describes the mobility more comprehensive from a worm spreading point of view because the carrying-spreading propagation in VANETs is characterized.

In this paper, the modeling methods for worm spreading on VANETs are covered by categorizing them into the analytical method, data-driven method, and simulator based method. Instead of only following one classification, we remark the mobility models used for worm spreading analysis on VANETs according to all the three classifications mentioned above. The details of mobility models and their classification can be found in Table 2.3. We

group the factors of the worm spreading into mobility factors, communication factors, and worm infection factors. Table 2.4 shows the factors covered by each model. The symbol $\surd$ indicates the value of the individual vehicular node is used and symbol $\overline{\surd}$ stands for the average value.

Table 2.3: Summary of Discussed Models

Year	**Scenario**	**Topology**	**Type of Mobility Model**	**Simulator**	**Data Driven**	**Mobility Model**	**Channel Model**	**Worm Propagation Model**
2004 [68]	Highway	Static	Marco	N/A	No	Uniform traffic density model	Log-normal shadow fading model	SIR epidemic model on random graph with worm life time
2006 [78]	Highway	Static	Micro	N/A	No	Steady-state IDM	Velocity-dependent log-normal shadow fading model	Multi-hop broadcast with infection probability
2007 [66]	Highway	Dynamic	Marco	PARAMICS	No	N/A	N/A	Diffusion-reaction model
2010 [46]	Highway	Static	Micro	N/A	Yes	Car following model	Dual-slope piecewise linear model	Multi-hop broadcast
2014 [91]	Urban	Static and dynamic	Micro	VanetMobiSim and NS2	No	IDM	Log-normal shadow fading model with MAC mechanism	Multi-hop broadcast with infection probability
2015 [89]	Mixed	Dynamic	Marco	MMTS	Yes	N/A	N/A	Broadcastcarrier worm propagation model
2016 [8]	Urban	Dynamic	NA	VEINS	No	N/A	N/A	Multi-hop broadcast

Table 2.4: Factors being Considered in Models

Years	Mobility Factors				Communication Factors						Worm Infection Factors		
	Traffic Density	Velocity	Interval	Lanes	Distance between nodes	Path loss effect	Fading effect	Comm-unication range	Packet Collision	Hops per second	Time required for self-copy	Virus strength	Worm life time
2004 [68]	√	$\overline{\surd}$	$\overline{\surd}$	√	$\overline{\surd}$	√	√	-	-	-	-	-	√
2006 [78]	√	$\overline{\surd}$	$\overline{\surd}$	-	-	√	√	-	-	-	-	√	-
2007 [66]	-	$\overline{\surd}$	-	-	-	-	-	√	-	√	-	-	-
2010 [46]	-	√	√	-	-	√	√	-	√	-	-	-	-
2014 [91]	-	√	√	√	√	√	√	-	√	-	-	√	-
2015 [89]	-	$\overline{\surd}$	$\overline{\surd}$	-	-	-	-	√	-	-	√	√	-
2016 [8]	√	√	-	-	-	-	-	√	-	-	√	-	-

2.3.2 Analytical Method

Analytical models are characterized by scenario assumptions, which means the model does not need to be fed by real traffic data. Because the detailed traffic data is privacy related, sometimes it is not accessible to researchers. In such case, analytical methods provide a possible way to implement a data-free analysis of worm spreading behaviors on VANETs.

Khayam and Radha proposed an active worm spreading analysis method by applying a SIR (Susceptible-Infected-Recovery) epidemic model on a VANET topology represented by a geometric random graph [68]. The geometric random graph is characterized by a macroscopic mobility model and a channel model. Instead of describing the mobility of the individual vehicle, the approach for static ad hoc network study was employed for building the VANET topology in [68]. They assume the vehicular nodes on a rectangular road section follow the uniform distribution. The VANETs connectivity with different traffic volume can be represented by the density of vehicular nodes, so the low-density and congested scenarios could be studied.

The influence of average effective distance on the speed of worm spreading on VANETs was investigated in [68]. The effective distance proposed by [64] is a distance factor characterized by average vehicle velocity, average time interval between two vehicles, and the number of lanes. The mathematic expression of effective distance is given in Equation 2.1 [64], where v_a is the average vehicle velocity, τ is the average time interval, and L is the number of lanes.

$$d_a = \frac{v_a \tau}{L} \tag{2.1}$$

Based on the average effective distance, the log-normal shadow fading model was employed to describe the probability of the communication link existence between two vehicular nodes. There are two components in the standard log-normal shadow fading model, one is a deterministic distance dependent variable, and the other is a normal random variable describing the fading effect [39]. The average effective distance and the transmission and receiving power are used to define the distance dependent component. Then the average probability of a communication link between two vehicular nodes can be represented by the average effective distance and a predefined receiver sensitivity. The mathematic expression of link probability is shown in Equation 2.2, where β_{th} and σ are the receiver characteristics and fading factor, respectively.

$$p_{link} = \frac{1}{2} + \frac{1}{2} erf\left(\frac{\beta_{th} - \alpha 10 log\left(v_a \tau / L\right)}{\sqrt{2}\sigma}\right) \tag{2.2}$$

The communication probability between vehicular nodes characterized by the average effective distance is further used for building up the geometric random graph. Then the influence of traffic density on the speed of worm spreading was analyzed by the average degree of the geometric random graph. Because Khayam and Radha [68] assumed the vehicles followed the same distribution in low-density and congested scenarios, the same equation was used for describing both low-density and congested scenarios. The mathematic expression of the average degree of each vehicular node is given by Equation 2.3 [68].

$$d_{avg} = \frac{1}{2}(\rho_a - 1)\left(1 + erf\left(\frac{\beta_{th} - \alpha 10log(v_a\tau/L)}{\sqrt{2}\sigma}\right)\right) \tag{2.3}$$

The simulation results in [68] agreed on their analytical conclusion that for both low-density and congested scenarios the average degree increased along with the increase of traffic density.

In [68], the worm spreading was studied based on the average node degree characterized by the average effective distance. The SIR epidemic model was employed to model worm spreading on the geometric random graph. The preemptive and interactive patching were used as two countermeasures for curing the infected vehicular nodes. They assumed an infected node only could be infectious within the worm lifetime, so the infectious period was introduced to characterize the worm spreading. The speed factor defined by the ratio of newly infected nodes to the total nodes was used as the indicator for the speed of worm spreading. The simulation results in [68] agreed with their analytical results that the worm spreading was more rapidly in congested traffic than low-density traffic due to the high average degree of each node. The preemptive patching could not curb the worm outbreak in both low-density and congested scenarios.

The worm spreading study in [68] is based on the VANET topology built by the traffic model and the channel model. The assumption of traffic distribution limits the applicable scenarios and the accuracy of this method. The VANET was modeled as a random graph based on the existence probability of the communication link. However, the distance between vehicle was derived from the traffic distribution. In the real case, the vehicle distribution cannot follow the uniform distribution, because the traffic density and distribution are dependent of the traffic condition. Moreover, the study proposed in [68] did not characterize the dynamic VANET topology along with time. Although the random graph provides a way to analyze the topology in a random style, only the radio communication related random factors are taken into account. However, the most significant feature of VANETs is dynamic topology which is contributed by the vehicle mobility and the traffic density. Because of the lack of characterizing vehicle movement, the VANET topology built by the method in [68] is a static VANET topology based on the instant positions of vehicular

nodes. For the same reason, the worm propagation caused by physical carrying-spreading cannot be described by this method.

A well know vehicle mobility model, Intelligent Driver Model (IDM), was employed to build the VANET topology for worm spreading studies. The IDM balances the complexity of microscopic mobility model and the highly abstract of macroscopic mobility model. Details about IDM can be found in [87]. The static-state IDM shown by equations 2.4 and 2.5 was used for describing the vehicle mobility on highway scenarios by Nekovee [78] and Wang et al. [91]. In static-state IDM, the traffic flow of one lane is characterized by the mean equilibrium gap S between two adjacent vehicles, average velocity V, average vehicle density ρ, desired velocity V_0, and minimum bumper-to-bumper distance S_0. The static-state IDM is based on the reality that the distance between a vehicle and its preceding vehicle is velocity dependent. V_0 and S_0 are predefined constant parameters. Compared with the mobility model in [68], traffic density ρ is also used in IDM but, instead of feeding as an input parameter, the vehicle density here is derived from average distance S and car length l. The expression of traffic density in IDM can be found in Equation 2.5

$$S(V) = (S_0 + VT)\left[1 - \frac{V}{V_0}\right]^{1/2} \tag{2.4}$$

$$\rho = \frac{1}{l + S} \tag{2.5}$$

Although IDM was used in both [78] and [91], their channel models were different. A velocity dependent shadow-fading model was proposed in [78]. This proposed channel model was also developed from standard log-normal shadow fading model, but the relative speed between two cars in the lanes of opposite direction was considered by modifying the model with a velocity-dependent deterministic component. The equation of link probability proposed in [78] is shown in Equation 2.6, where the r_{ij} and v_{ij} are the distance and relative velocity between vehicular nodes i and j.

$$p_{ij} = \frac{1}{2} + \frac{1}{2} erf\left(\frac{\beta_{th} - \tilde{\beta}_1(r_{ij}, v_{ij})}{\sqrt{2}\sigma}\right) \tag{2.6}$$

With the VANET topology built by the static-state IDM and velocity dependent shadow-fading model, the worm propagation was studied by the average link probability between two vehicular nodes.

In [78], broadcasting was assumed as the worm target finding scheme, which means all the nodes within the effective communication range were vulnerable to infection. To

study the impact of the worm characteristics on the worm spreading, an infection rate λ was designed to represent the worm strength. The infection rate is defined by the ratio of infected nodes to all the nodes receiving the worm packets. For one infectious node and a certain number of reachable nodes, a higher infection rate means the higher worm strength, so a greater number of reachable nodes would be infected at the end of worm propagation process.

The IEEE 802.11p was assumed to be the V2V multi-hop communication protocol in [91], so the effective communication range could be estimated. To find out the link probability between two vehicular nodes, the log-normal shadow fading model was used as the channel model [99]. Then the estimated effective communication range was used as the distance to feed the channel model. To incorporate the 802.11p protocol, the underlying MAC mechanism of the communication channel was taken into account [6]. The listen-before-talk (LBT) mechanism was designed in MAC layer to avoid collision and interference. The LBT mechanism allows only one node to send or receive the message at each instant. According to the LBT rule, each vehicle has an equal chance to occupy communication channel. In the simulation of [90], an ID ring was designed to imitate the occupation of the communication channel [25]. All the vehicle IDs on the ring would be selected to be transmission candidate by turns. The first vehicle ID was random selected and assigned the "token". The "token" would pass to the neighboring ID in the clockwise direction. This transmission candidate selection rule guarantees only the vehicle possessing "token" sends the message at each moment. The candidate selection process repeats until all the vehicles in the ring have sent a message once.

The worm spreading model developed in [78] was evaluated in a 10 kilometers bidirectional single-lane highway scenario. The locations of the vehicles on the road are assumed to follow the normal distribution [2]. The average speed is assign to all vehicular nodes. The VANET connectivity is evaluated by the size of the largest cluster. In the worst case, an infected node could infect all the reachable nodes ($\lambda = 1$), so the largest connected cluster is used to estimate the upper bound of the number of nodes can be infected. It is known that the connectivity of vehicular nodes in VANETs highly relies on the traffic conditions, so the simulation is carried out with the different average speed. The simulation results in [78] were given by the varying of infection ratio along time-step. According to their simulation results, the authors concluded that the average vehicle velocity strongly influenced the speed of worm spreading [12]. The worm spreading speed reached the peak in high traffic density condition, but it was much slower than the exponential worm spreading rate known on the Internet.

The simulation of [20] was carried on a 1 Km^2 urban topology with bi-directional two-lane roads generated by VanetMobiSim. The static and dynamic VANETs topologies

were considered respectively. Because the study of the worm spreading on a dynamic topology was a simulator-based study, we leave it for Section 2.3.4. In the static topology, Monte Carlo method was used to imitate the worm spreading on the VANET built by the proposed model. The velocity of each vehicle was assigned randomly within a speed range so that the distance can be derived from the mode. At each Monte Carlo iteration, a set of vehicles were randomly selected as the first infected vehicular nodes, and then the worm spread based on the proposed worm spreading model [91]. The results were given by the average value of more than 100 Monte Carlo iterations. Based on the simulation results, the authors concluded that worm spreading was directly proportional to transmission range, vehicle velocity, and traffic density in the static topology. The author also found that the MAC mechanism slowed down the worm propagation process.

Compared with the method in [68], the MAC mechanism and relative velocity were considered by the worm spreading studies proposed in [78] and [91], respectively. It is notable that the static IDM is employed in both [78] and [91]. Because IDM could characterize the vehicle mobility by road section, it provides the flexibility to describe the traffic flow of each road section in the interested area. Furthermore, IDM estimates the bumper-to-bumper distance with the given traffic density and average speed of the current road section. The bumper-to-bumper distance can be used for the channel model to describe the VANET topology. However, the methods proposed in [78] and [91] are limited by their assumptions and the static topology. In [78], the average speed was assign to all vehicles on the road, so the vehicles follow the uniform distribution. As we pointed out previously, the uniform distribution assumption may not hold in real traffic flow. In [91], a random speed within a predefined range was assigned to each vehicular node to find the distance between vehicular nodes. But, in real traffic flow, the vehicle speed is dependent of the preceding vehicle and traffic density. Another limitation of these two methods is the absence of consideration for the vehicle movement. The topologies built by static IDM model with the average speed can be seen as a series of snapshots of the dynamic VANETs, so the relation between the vehicle positions and time cannot be revealed. As a result, the models for worm spreading on VANETs proposed in [78] and [91] cannot describe the worm propagation by physical carrying-spreading.

2.3.3 Data-driven Method

Most analytical models rely on the assumption of vehicle distribution to estimate the distance between vehicles. For example, the uniform distribution and normal distribution were employed in the mobility models proposed in [68] and [78]. However, the assumed distribution model may not hold in real world [46]. Serval data-driven models were proposed

to avoid the unrealistic assumption for the study of the worm spreading on VANETs.

Cheng and Shakya [46] utilized an empirical method to develop the model for worm spreading on road networks. The extended car following model proposed in [95] was employed to describe the traffic flow. The key factors in the extended car following model are the vehicle speed, bumper-to-bumper distance, and inter-arrival time. In [46], the parameters used in the model were found by a statistic analysis on the real traffic data. The real data under different traffic volume were collected to determine the vehicle distribution according to various traffic conditions. Kolmogorov-Smirnov test (K-S test) was first applied to obtain the empirical distribution, and then the smallest D-statistic analysis was implemented to find the value for each parameter. They concluded that the exponential distribution should be used for the extreme light traffic and generalized extreme value distribution should be used for the moderate and heavy traffic.

The communication model used in [46] was also derived by an empirical method. A dual-slope piecewise-linear channel model developed by the authors in [45] was used as channel model to estimate the communication range. The dual-slope piecewise-linear channel model is characterized by path loss exponents γ_1, γ_2 and the standard deviations within or out of the critical distance. In their study, all the parameters in dual-slope piecewise-linear channel model were specified based the data sets acquired in the same suburban scenario. Broadcasting and message flooding were used as worm propagation pattern. An infected vehicular node would broadcast the infectious message, and all the nodes within its communication range would be infected. As a result, the infection message would flood the reachable nodes in the network.

The simulation in [46] was also carried out on a 10 kilometers highway corridor. The vehicle distributions found by the previous empirical method was used to determine the location of each vehicular node. The infection process was triggered by a single infectious node [20]. One vehicular node was randomly selected as the initial infectious node starting to broadcast the infectious message. All the neighboring vehicular nodes within the effective communication range could be infected and start to spread the worm by broadcasting. The ratio of the infected vehicles to the total susceptible vehicles was defined as the indicator to analyze the worm spreading along with time. The data under different traffic conditions was applied to evaluate the how the traffic density would impact the worm spreading. The simulation results in [46] show the ratio of the infected vehicle reaches the peak in rush hours and the speed of worm spreading depends on the traffic density. It is notable that the approach of connectivity estimation by the vehicle distribution is also a frozen-network style approach, so the time span can be used to evaluate the worm spreading is very limited [46]. In their simulations, the worm spreading process was assessed within 2000 ms, because the topology would change significantly when the

time span was increased. Therefore, the large-scale spatiotemporal spreading of the worm cannot be described by this method, because the carrying-spreading propagation cannot be taken into account.

Instead of a combination of the mobility model, communication channel model, and worm epidemic model, Oscar Trullols-Cruces [89] developed an integrated worm epidemic model for VANETs exclusively. The broadcast-carrier worm propagation model proposed in [89] is characterized by the vehicle mobility, communication and worm characteristic parameters in nature. The model is driven by the parameters of each road section, such as average vehicle speed and average interval. These parameters can be acquired by statistic traffic data of the road section. The worm is characterized by carrier latency and penetration which are dependent on the size of worm body and the worm strength, respectively. The communication capability of a vehicular node is abstracted by the communication range. The carrier latency, penetration ratio and communication range are constant inputs feeding the proposed model, where carrier latency is the time required for worm self-copying, penetration ratio is the percentage of susceptible vehicles to total vehicles, and communication range is the maximum distance between two vehicles being capable of setting up a communication link. The mathematic expression of the broadcast-carrier worm propagation model proposed in [89] is given in Equation 2.7.

$$s_i = v_i + \frac{R}{\tau} exp\left[-\left(\frac{3a_i v_i}{\rho R}\right)^2\right] \tag{2.7}$$

where s_i, a_i, and v_i are the worm propagation velocity, the average interval time, and the average vehicle velocity of the road i, respectively. τ is the carrier latency, ρ is the penetration ration, and R is the effective communication range.

Compared with the previous models, the broadcast-carrier worm propagation model describes the node connectivity based on the traffic on each road section. Besides, it further characterized the worm propagation by the time required for transferring worm body. To show the accuracy of the proposed model, the prediction result acquired by applying the proposed model was compared to statistic results generated by the multi-agent microscopic traffic simulator (MMTS).

The author argued that vehicular worms would be capable of spreading throughout a large area to infect a vast number of vehicles within a short period of time [89]. The author concluded that the high dynamic topology and V2V communication were the main reasons of the widely and rapidly worm spreading. Besides that, the author also discussed the danger level of initial worm locations. According to their simulation, the danger level

would increase when the initial worm location was closer to urban center or when it had higher local road traffic density.

The worm spreading on VANETs was also studied from an information diffusion aspect. Hoh and Gruteser [66] suggested a worm spreading analysis method by diffusion-reaction model. This study aimed at estimating the boundary of the area containing infected vehicular nodes so that the worm infection could be quarantined. By applying the diffusion-reaction model, the worm propagation was studied in a macroscopic point of view. The worm spreading process was modeled as an isotropic dispersal process starting from the origin of a polar coordinate. The worm diffusion process is characterized by a constant diffusivity and growth rate. The growth rate depends on population density, communication range, and node velocity. According to diffusion-reaction model, the frontal wave of the diffusion process would diffuse at a constant speed as long as the node density and node velocity is constant. However, the vehicle density and average speed may vary for individual road section. As a result, the diffusivity and growth rate is hard to estimate. To overcome this challenge, the upper bound of propagation velocity was used as the worst case of the worm spreading to determine the boundary of the infection area. The upper bound of propagation velocity was derived from communication range and the average vehicle speed of the current road section. The simulation was carried out in [66] based on the traffic data generated by PARAMICS transport system simulator. A variety of communication ranges were implemented in the simulation with the respect of different path loss of V2V communication. The simulation results were presented by the increasing radius of the frontal wave of worm propagation along with the elapsing time since worm propagation started. The authors concluded that the proposed model could be used for worm spreading boundary prediction, and a worm could widely spread in highway scenario even if only a small ratio of vehicles were susceptible. The diffusion-reaction model in [66] highly relies on the accuracy of the location estimation of the first infectious node, and it only works with single worm source scenario. For example, this estimation model cannot work when a number of vehicular nodes are infected and randomly distributed in the road network.

The data-driven models can be calibrated by the real data, so the traffic flow can be described more realistic. But the accuracy of the model is dependent of the traffic data. Both of the studies in [46] and [66] tried to investigate the worm spreading along with time. However, due to the limitation of the mobility model, only a short period of propagation time could be considered in [46]. The study proposed in [66] considered not only the worm spread by V2V communication, but also the carrying-spreading caused by vehicle mobility. The conclusion in [66] shows that the vehicle movement has a significant contribution of worm spreading, especially in low-density traffic scenarios. However, the method proposed

in [66] only works with single infectious source and it is effective only if the initial location can be estimated accurately. Although the vehicular nodes could be used for monitoring and target tracking [83] [28], the worm may be spatially distributed on the VANETs with multiple initial locations, so the applicable scenarios of the method proposed in [66] are limited.

2.3.4 Simulator Based Method

The mathematical expressions of the vehicle mobility and the communication model provide a way to study the worm spreading on a vehicular network analytically. However, the static models describe vehicular networks in a frozen-network style. The instantaneous position of each vehicle is used to define the network topology. It is known that describing the varying topology along with time is challenging. To overcome this limit, vehicular network simulators have been introduced to analyze the worm spreading. By taking advantage of using simulators, researchers could focus on the study of the worm spreading process but building the underlying VANET topology.

Along with the blooming of VANETs studies, many simulators have been developed and widely utilized. They can be categorized into vehicle mobility simulators, network simulators, and integrated simulators. Vehicle mobility simulators generate individual vehicle movement based on the input map data, such as Paramics and SUMO. Network simulators imitate communication protocols and radio propagation based on the distance between nodes and predefined environment parameters, such as NS-2, NS-3, OPNET, and OMNeT++. The integrated simulators combine the function of a vehicle mobility simulator and a network simulator and provide a more user-friendly interface, such as GrooveNet, TraNS, and VEINS.

Wang et al. [91] presented a simulator-based study as a supplement to their analytical study to investigate worm propagation on dynamic topology. The worm spreading on a dynamic topology was implemented by connecting VanetMobiSim and NS2. The simulation results were presented by the ratio of the infected vehicles along with the time of worm spreading. According to their simulator-based study, the authors concluded that transmission range, vehicle velocity, and traffic density were directly proportional to the speed of worm spreading, which agreed on the conclusion for static topology. Besides, the simulator helped to reveal the relation between the vehicle speed and the worm outbreak such that vehicle speed reduced the minimum infection rate required for leading to the worm outbreak throughout the whole network.

A pure simulator-based study was proposed by Basaras et al. [8]. They investigated

the worm spreading process and suggested a containment approach to avoid susceptible vehicles being infected. A blacklist containing the infectious vehicle IDs would be disseminated by special hardware in VANETs system. By doing so, the susceptible vehicles can identify the surrounding infectious vehicles and drop the packets from those vehicles. Instead of formulating the vehicle mobility and radio connectivity, an integrated VANETs simulator, VEINS, was used to analyze the worm spreading and to assess their proposed countermeasure. With the help of the simulator, the traffic flow of the interested area was generated by defining the movement of each vehicular node along with time. Then the corresponding communication behavior between vehicular nodes of each instant could be found. The worm spreading process was analyzed by traffic density, initial locations of infectious vehicular nodes, transmission delay, and virus strength. Transmission delay is the time required to complete the self-copy of worm body and virus strength refers to the ratio of vehicles that are susceptible to this worm. The simulation results in [8] show that the proportion of infected vehicles depends on the size of worm body and virus strength. More precisely, the longer time required for worm self-copying, the less ratio of nodes would be infected. The higher virus strength, the more vehicles would be infected.

The development of VANETs simulator provides a possible way to study the worm spreading by experiments. However, most of the simulators work by some integrated mobility and communication models. Not all the simulators provide the details about how those models are implemented, so it is hard to compare the simulation results and conclusion across simulators. Besides, the results of simulation based methods are highly dependent of the accuracy of the simulators. Sometimes the conclusions from simulation results could not be proved by mathematic derivation.

2.4 Worm Containment Strategy

The existing studies show that the speed of the worm spreading on VANETs is dependent of vehicle mobility, effective communication range, and worm characteristics. Both analytically and simulation studies indicate that a significant fraction of vehicular nodes would be widely and rapidly infected without a proper countermeasure. However, the worm spreading countermeasure for conventional computer networks cannot be applied to VANETs environment due to the reasons discussed in Section 2.2.

As an important part of worm propagation studies, some containment strategies have been proposed to restrain worm spreading on VANETs. The existing counter methods and their evaluation matrix are listed in Table 2.5 and discussed in this section.

Table 2.5: Countermeasures and Evaluation Matrix

Year	Counter Strategy	Cure Dissemination Method	Evaluation Matrix
2004 [68]	Preemptive patching and interactive patching	V2V	Ratio of infected vehicles along with time
2006 [78]	Preemptive patching and interactive patching	I2V and V2V	Ratio of infected vehicles along with time step
2010 [46]	Interactive patching	I2V & V2V	Ratio of infected vehicles along with time Traffic density, Broadcast timeout, Patching/without patching
2015 [89]	Smart cellular patching	I2V	Initial location, Carrier latency, Traffic density, Virus Strength
2016 [8]	Broadcast BL, PIV	I2V and V2V	Ratio of infected vehicles along with time, Infection delay

2.4.1 Preemptive and Interactive Immunization

Preemptive immunization and interactive immunization are two main countermeasures for worm prorogation on VANETs. The immunization based methods are the first and commonly used counter strategy for combating worm spreading on VANETs. The immunization based methods are developed based on the nature of the SIR epidemic model. In SIR model, each node has three status, susceptible, infected, recovery. All the nodes in the system are susceptible in the initial stage. After the worm spreading process triggered in the system, the nodes impacted by infection turn to infected nodes. An infected node becomes recovery after receiving a vaccine injection. A recovery node is not susceptible to worm epidemic. In computer networks and VANETs environment, sending the curing patch to an infected node is considered as vaccine injection process corresponding to SIR model.

The preemptive immunization refers to patch the susceptible nodes before a worm spreading on the network, so it is only effective for the known worms. When the susceptible nodes are numerous, it is impossible to patch all of them at the same time. As a result, the worm still could propagate and infect those unpatched nodes. The challenge of preemptive immunization is to select a proper set of nodes to immune such that the total number of

infected nodes is minimized or the delay of worm epidemic outbreak is maximized. The existing studies show that the preemptive immunization strategy is unlikely effective in VANETs. Because of the high dynamic network topology and the vehicular node mobility, it is hard to find a set of vehicular nodes that could effectively block the worm propagation. Besides, with the broadcasting and multi-hop communication, a worm could easily bypass an immune vehicular node by an alternative path provided by other nearby nodes.

When a new type of worm appears, the interactive immunization is applied. In interactive immunization scenarios, the curing patch and the worm are spreading in the network simultaneously. The objectives of interactive immunization are preventing the uninfected nodes from infection and recovering the infected nodes as soon as possible. Different patching strategies for interactive immunization were proposed such as patching by the cellular base stations, by the V2V communication, and by a mixed strategy of both I2V and V2V communication.

The effectiveness of preemptive immunization and interactive immunization have been assessed in the existing studies. The spread factor defined by the ratio of infected vehicular nodes to the total susceptible nodes is commonly used as the indicator to evaluate how widely the worm spread. The patching rate defined by the ratio of the vesicular nodes being patched initially is used to illustrate the effectiveness of preemptive immunization [64] [78]. The simulation results in [64] show that the patching rate significantly impacts on the worm spreading. Compared with preemptive patching, interactive patching can effectively mitigate the worm outbreak in both high and low traffic density scenarios. Similarly, the effectiveness of preemptive immunization and interactive patching were also assessed by simulation in [78]. In their preemptive immunization scenario, all susceptible nodes without immunization turned to infected even if 30% nodes were preemptive immunized. In contrast, the interactive immunization was shown to be effective with various patching rate. They also revealed that a higher vehicle speed could accelerate the patch diffusion.

Besides the patching rate, the area where the curing patch to be released first was discussed in the interactive immunization strategy. A cellular-based smart patching method was suggested in [89] to counter the worm spreading in VANETs. According to this strategy, the curing patch is sent to the vehicular nodes by existing 3G/4G base stations. The patched vehicular nodes do not forward the curing patch to other vehicles by V2V communication. Instead of patching all the vehicles on the road, the cellular-based smart patching method first estimates the infected region according to the time span since the first infection appeared, and then the curing path is released only in the infected area. Although this region selective patching reduces the number of vehicles for immunization, its performance is limited by the accuracy of infected area estimation.

The interactive immunization strategy has been shown to be effective in the existing studies. However, it is unlikely to be a perfect countermeasure for worm spreading. Because a curing patch has to be developed before it is released to vehicular nodes, there would be a delay before the curing patch is disseminated on the network. According to the conclusions of worms spreading behavior in the existing studies, a significant ratio of vehicular nodes would be infected by V2V communication during the delay of curing patch development. So, even though the interactive immunization has been shown its effectiveness for countering worm spreading, it is very challenging in practice.

2.4.2 Blacklist Isolation

In addition to immunization based approaches, a blacklist based worm containment strategy was suggested by Basaras et al. [8]. The authors argued that the delay caused by developing the patch for a new worm would be uncertain, so they considered the anti-spreading and curing process separately. According to their strategy, a blacklist containing the IDs of infected vehicular nodes is disseminated by a type of special Road Side Units (RSUs). The blacklist is designed to be shared between vehicles by V2V communication. Compared with immunization, the blacklist cannot cure the infected nodes but only prevent the susceptible nodes from being infected. However, the blacklist isolation is supposed to respond more rapidly than the immunization approaches, because it does not require the curing patch. The isolation would be effective as long as the infectious vesicular nodes are identified before the worm widely spread.

Because the worm spreading and immune recovery process are considered separately in [8], the worm outbreak was expected to be delayed by the dissemination of the blacklist among the susceptible vehicular nodes. The simulation results in [8] show that the ratio of infected vehicles depends on the size of worm body and virus strength. The traffic density and initial location of infectious vehicles also impact the worm spreading, but, because the spread of the blacklist is directly proportional to the speed of worms spreading, the worm can be inhibited even if in high connectivity scenarios.

2.5 Conclusions and Open Issues

Vehicular nodes containing high-value and privacy information are vulnerable to malware attack. However, compared with conventional computer nodes, a vehicular node is more susceptible to malware attack, such as worms, because it is multi-interface accessible.

Table 2.6: Factors and Their Influence on the Speedy of Worm Spreading

Spreading Speed Factor	**Direct Proportional**	**Inversely Proportional**
Traffic density	√	
Distance between nodes		√
Time interval		√
Vehicle velocity		√
Communication range	√	
Average bandwidth	√	
Path loss		√
Fading effect		√
Size of worm body		√
Virus strength	√	

Worm epidemic spreading on VANETs behaves differently from the conventional computer networks. The models developed for computer networks cannot work in VANETs environment because of the high dynamic topology and short-range multi-hop communication between vehicular nodes. The analytical studies and simulation results show that the worm epidemic spreads widely and rapidly without a proper countermeasure.

The speed of worm spreading is governed by traffic conditions, communication capability, and worm characteristics, such as traffic density, vehicle velocity, moving path, effective communication range, initial infection location, and worm strength. The influence of each factor on the speed of worm spreading on VANETs is summarized in Table 2.6.

Although a variety of strategies have been proposed for countering worm spreading on VANETs, their effectiveness is limited by incomplete worms libraries, long response time and the lack of nodes selection strategy for prior patching, etc.. The blacklist based counter strategy suggests a possible way to overcome the drawbacks of immunization patching. However, the special RSUs deployment strategies to effectively disseminate the blacklist and the locations of base stations to broadcast the curing patch should be further discussed. Due to the vehicle mobility, a worm propagated by physical carrying-spreading may cause a spatial widely dissemination, but the study on worm spreading resulting from this propagation approach is insufficient. Therefore, more efforts are expected in both the special RSUs deployment issues and the worm spreading caused by physical carrying and spreading.

Chapter 3

Worm Containment on VANETs

Worm spreading among vehicular nodes by V2V communication and carrying-spreading will cause security issues and plague the development of VANETs. The existing studies of the worm spreading on VANETs mainly focus on the worm propagation. Although some countermeasures have been proposed, most of them rely on 3G/4G cellular base stations or some special RSUs. The discussion on the deployment of special RSUs to inhibit worm spreading is very limited. Our study is motivated by this reality, and our objectives are to study the worm propagation on VANETs and to develop a method suggesting a deployment strategy for special RSUs.

Worm combating methods discussed in Section 2.4 are mainly infrastructure centered and patching based. The infrastructures could be some special RSUs or 3G/4G cellular base stations. For both RSUs and base stations, they have similar characteristics such as immobility, limited communication capacity, limited communication range and backhaul network required. Patching method refers to disseminate a special patch that can immunize a susceptible vehicle or cure an infected vehicle. Preemptive and interactive patching are two patching strategies. The preemptive patching is limited by the varying VANETs topology and unpredictable worm characteristics. The interactive patching is effective under various traffic conditions and scenarios, which has been proved by the existing studies. However, based on the worm spreading behavior discussed in Section 2.2, it is very challenging to develop a curing patch for a new worm before the worm outbreak. Compared with the patching based worm countering approach, the blacklist based methods consider the worm inhibiting and curing process separately. The blacklist based method would take effect as long as the vehicles conducting malicious behavior are identified. Therefore, we argue that the blacklist based worm countering method would be more feasible to inhibit worm spreading before the worm outbreak.

3.1 Worm Spreading on VANETs

Worm spreading process on VANETs is significantly different to the conventional computer networks. The dynamic topology of VANETs limits the scanning and hit-list based target finding approaches. The multi-hop communication limits one-hop propagation distance. However, the vehicle mobility and the message broadcasting employed by V2V communication enable an infectious vehicular node to infect a significant number of neighboring vehicular nodes in a wide time-spatial span. In this section, the characteristics and influential factors of different VANETs worm propagation approaches are discussed.

3.1.1 Approach of Worm Spreading

The V2V multi-hop communication is a fundamental feature of VANETs. From a worm spreading aspect, an infectious node sends worm body to its reachable neighboring nodes by V2V communication. This infection procedure is repeatedly for ever newly infected node. So, in high traffic density scenarios, the speed of worm spreading is rapidly, and the number of infected nodes increases exponentially. The existing study results show that for a particular type of worms (the worm strength is constant), the speed of worm spreading by V2V multi-hop communication depends on the traffic density, effective communication range, and the average available bandwidth. Because the worm propagation by multi-hop communication is rapid, the displacement of each vehicle during this tiny time slot is negligible. As a result, the relative distance between any two vehicular nodes could be seen as static. Therefore, a snapshot of the dynamic VANETs topology can be used to analyze the multi-hop based worm spreading.

The limits of multi-hop worm spreading are evident. First, when the traffic density is low, or the distance between two nodes is further than one-hop communication range, the propagation will fail. Second, the multi-hop spreading hardly leads to a spatial widely worm spreading due to the limited number of reachable nodes within one cluster of vehicular nodes in VANETs environment. Therefore, we conclude that the V2V multi-hop communication approach contributes to a rapidly local worm spreading.

The worm spreading process on VANETs is unique from the conventional computer networks because of vehicle mobility. Compared with the worm propagation by V2V multi-hop communication, the speed of worm propagated by a vehicle physically carrying-spreading is less rapidly. The propagation speed depends on the velocity and route of the vehicle. The spreading trend depends on the route and destination of the vehicle carrying the worm body. Note that, when a vehicle carrying the worm is moving on the road, it is infectious for every vehicular node passing by it. Although this high dynamic mobility

limits the possibility to attack a certain target, it significantly increases the possibility to infect a large number of vehicular nodes. More importantly, from a large time-span aspect, the worm would be inevitable spatial widely spread even in the low traffic density scenarios.

In fact, an infectious vehicle spreads worm by both the multi-hop communication and carrying-spreading simultaneously. In high traffic density scenarios, the worm spreading mainly relies on multi-hop communication, the speed of worm propagation is much faster than the velocity of the infectious vehicular node. When an infectious vehicle in a low traffic density scenario, the worm spreading mainly depends on the vehicle mobility. We define a vehicular cluster is a group of vehicular nodes reachable to each other by V2V multi-hop communication. Then the worm spreading process can be rephrased as following. When an infectious node appears within a vehicular cluster, all the nodes in this cluster will be infected at the first time by multi-hop communication. Then the newly infected vehicles in the cluster border will carry the worm to other vehicular clusters.

From a large spatial-temporal span aspect, the traffic density is diverse from area to area and varying from time to time. So it is challenging to describe the worm propagation when the large spatial-temporal span is taken into account. However, for both multi-hop based and physical carrying based worm spreading, the worm propagation paths depend on the vehicular nodes, and the vehicular nodes can only move on accessible road sections. As a result, the possible worm propagation paths are restricted to the underlying road network.

3.1.2 Worm Propagation on Road Networks

Instead of modeling the worm spreading on a dynamic ad hoc network topology, we analyze the worm spreading problem of VANETs by the underlying road networks. According to the existing studies, the speed of worm spreading on VANETs depends on worm strength, traffic density, and vehicle mobility. The worm propagates via V2V wireless communication and vehicle physically carrying-spreading. Both of this propagation approach rely on vehicular nodes, and the vehicular nodes are constrained by the road topologies. In the perfect condition, the vehicular nodes are uniformly distributed on each road section, any pair of vehicular nodes is reachable to each other by V2V multi-hop communication. In this case, the VANETs topology is roughly the same as road topology.

The perfect condition is the best case regarding the connectivity of vehicular nodes and the worst case of worm propagation. According to the conclusion from existing studies, in the perfect condition, the worm would outbreak almost instantly. When the worm could be

inhibited in the worst case, it could be curbed in all other cases, since the worm spreading will not be more violent than the worst case. Therefore, it is reasonable to study the worm spreading with the assumption of perfect condition.

It is known that the high dynamic VANET topology is caused by the vehicle mobility and the limited wireless communication range. However, the underlying road network is static. The road network topology could be seen as the topology of VANET with perfect connectivity, so it is possible to study worm spreading by underlying road networks instead of the dynamic VANETs topology. Therefore, from a macroscopic aspect, the worm is spread on the road network no matter the propagation approaches.

A road network in our definition is a network based on natural connectivity of the intersections and roads linking two adjacent intersections, which is known as the prime approach. The advantages of studying the worm spreading on road network topology are evident. First, it is avoided that modeling the dynamic VANET topology. Secondly, it is possible to apply the existing countermeasures for worm spreading on conventional computer networks to curb the worm spreading on road networks. Thirdly, by considering worm spreading on a road network, the worm propagation by V2V communication and physical carrying-spreading can be seen as one integrated propagation process because both of them propagate along the same road network.

A worm spreading process on road networks can be considered as a spanning tree that is rooted at the initial infectious node and spans along connected road sections. The propagation process stops at a road section when there is neither enough vehicular nodes to set up a V2V multi-hop communication link between two intersections nor any infected vehicle carrying the worm pass through the road section. Note that each path originating from the initial infectious node is independent, because the probability that a worm can be delivered through a road section is independent. Therefore, instead of considering the propagation process as a spanning tree process, we consider the worm propagation as a set of spanning paths starting from the same source node.

3.2 Worm Containment by Blacklist

We propose a Malicious Vehicle Screening Unit (MVSU) based countermeasure to inhibit the worm spreading on VANETs. The MVSUs is a particular type of RUSs which are capable of sniffing the malicious behavior and broadcasting the blacklist. The malicious behavior sniffing refers to receive the packets sent by every passing by vehicle and identify the malicious behavior. The blacklist contains an ID list of the vehicle conducting malicious behavior. The malicious behavior could be any activity impacting the normal functions

of other vehicular nodes or the VANETs system. In our research, the malicious behavior is defined as the act of actively worm spreading. The blacklist refers to the ID list of infectious vehicular nodes. The worm sniffing and blacklist immunization, as two main parts of the blacklist based worm containment strategy, are explained separately.

3.2.1 Worm Sniffing

We consider the worm containment and worm curing processes separately because of the concerns of the delay caused by curing patch development. In our research, we focus on the strategy to reduce the ratio of vehicles being infected by worm spreading. In our proposed method, the blacklist is used to avoid the susceptible nodes communicate with the infectious nodes. The vehicular nodes have to update the blacklist as soon as possible. A susceptible vehicle becomes immune when its blacklist is updated. Therefore the effectiveness of this proposed method mainly relies on the percentage of vehicles could be immunized before they are exposed to infectious nodes.

We defined an infectious vehicular node is "identified" when its ID is listed in the blacklist and "unidentified" when its ID does not appear in the blacklist. A vehicle is susceptible if it either failed to update its blacklist before entering a particular area nor there is an unidentified infectious node within the area. Therefore the locations of MVSUs should be capable of maximizing the probabilities of both sniffing the infectious vehicles and updating the blacklist of susceptible vehicles regardless the initial location and the mobility pattern of the vehicular nodes.

When an infectious vehicle appears, the worm starts self-spreading by V2V multi-hop communication, meanwhile the mobility of the infectious vehicles extend the infection areas that cannot access by straight forward V2V communication. For both V2V multi-hop communication and physical carrying-spreading propagation approaches, the worm is propagated along the road network and follows the propagation paths starting from the initial location of infectious nodes. So the locations with most probability to block the propagation paths are also the locations with most probability to sniff an infectious vehicle.

3.2.2 Blacklist Immunization

For an interested area, susceptible vehicles become immune to the worms within this area by updating its MVBL (Malicious Vehicle Black-list). The vehicles successfully updating their blacklist before entering a particular area will not be infected by the infectious nodes

within this area. After the immune vehicle entered a particular area, its blacklist could be shared by V2V communication, so all the vehicles within this area have a chance to acquire the update of MVBL. In contrast, the vehicles fail to receive the update of the MVBL from MVSUs are susceptible to the infection until they receive an update by V2V communication. According to the existing studies of the worm spreading on VANETs, it is only a matter of time that those susceptible vehicles finally become infected if there is not any blacklist update disseminated by V2V communication.

Since both the infectious and susceptible vehicles are moving along road networks, the goal of MVSU deployment strategy for worm sniffing and MVBL updating can be integrated and generalized from the vehicle movement point of view. The desired MVSU deployment strategy should be able to maximize the probability of a vehicle passes by an MVSU regardless the vehicle mobility and route selection preference.

Chapter 4

MVSU deployment problem

An MVSU is a special RSU that is capable of screening the worm propagation by sniffing the infectious vehicle and disseminating the blacklist containing the IDs of infectious vehicles. "Screening" means a special RSU can identify whether or not a vehicle has malicious behavior such as worm spreading. A susceptible vehicle can avoid communicating with an infectious vehicle by checking the internal blacklist containing infectious vehicle IDs.

The most challenge of curbing worm propagation by MVSUs is to find the proper locations for deployment. We define this problem as MVSU deployment problem on road networks. To elaborate our objective clearly, we study the scenario with following assumptions: The infectious vehicle could first appear at any intersection on a road network. Worms could propagate along every accessible road section and could be terminated at any road section. Due to the limited capability of sniffing and blacklist disseminating of each MVSU, we assume each MVSU is only capable of sniffing and disseminating blacklist to the vehicles moving in one direction. More precisely, one MVSU is required for a unidirectional road section, and two are needed for a bidirectional road section. Each MVSU can thoroughly sniff and communicate to the vehicles on the one directional road section such that an infectious vehicle will always be identified while passing by an MVSU and meanwhile all vehicles passing by an MVSU will receive the update of the internal blacklist. In another word, a susceptible vehicle will be immunized by passing by an MVSU and the infectious vehicles' IDs will be identified and added to the update of the blacklist. The higher ratio of vehicles on the road would pass by an MVSU the better worm spreading containment result could be expected. However, the number of total available MVSUs is limited, so a deployment strategy to maximize the effectiveness of worm containment should be investigated.

4.1 Worm Diffusion on Road Networks

Recall the discussion in Section 3.2, the locations to deploy MVSUs should be capable of covering the two goals, sniffing infectious vehicle and immunizing susceptible vehicles. The two goals are correlative to each other because the worm propagation process and the moving pattern of a susceptible vehicle can be represented by a simple diffusion process on a road network. In this section, the Independent Cascade (IC) Model is introduced to illustrate the conformities of the two goals. After that, the objective of MVSU deployment strategy is formulated by a diffusion process under IC model.

IC model is designed to describe the behavior of information propagation on networks. A digraph is built based on the topology of the network. The direction of each edge is defined by the reachability from one node to its neighbors. In IC model, after a message containing a piece of information originates at the source node, the message will be spread out by the following rules: each node holding the message has only one chance to select one out-edge to send the message. Each edge $\vec{e}_i$ has the probability $p_{\vec{e}_i}$ determining whether or not the message could successfully pass through. The message keeps propagating until either a failure occurs while the information passes through an edge or the message arrives at a node that already received it. IC model is a diffusion model which is independent of the propagation route selection preference. This characteristic perfectly suits our goals, since we are looking for an MVSU deployment strategy could effectively inhibit the worm spreading regardless the initial location and the vehicle mobility. The propagation probability of each edge $\vec{e}_i$ is the only one parameter in the IC model. As long as our two study objectives can be described by IC model, it is possible to study the worm spreading on VANETs by one formulated objective.

4.1.1 Worm Spreading by V2V Communication

The worm spreading in VANETs mainly relies on V2V multi-hop broadcasting. We employ IC model to represent the worm spreading by V2V communication. A road network can be represented as a graph by the prime approach. When an infectious node appears, its nearest intersection is used to represent its initial location approximately. Then each worm spreading process can be described as a diffusion process starting from an intersection. For a particular type of worms and standardized vehicular nodes, the worm strength and effective communication range are constant. The speed of worm diffusion by V2V communication only depends on the traffic density. According to the existing studies on worm spreading by V2V communication, the worm spreading process along a certain road section i can be expressed by a V2V delivery probability p_i. The physical implication

of p_i is that, for two neighboring intersections u and v connected by a road section i, p_i determines whether or not a V2V multi-hop communication link exists from intersection u to intersection v. Because we only consider the multi-hop based propagation, the speed of worm spreading is much greater than the velocity of a vehicle, so the position of vehicles could be considered as static. In this case, the V2V delivery probability p_i only depends on the traffic density.

As discussed in Section 3.1, the worm propagation can be represented as a set of spanning paths starting from the same source node. When a road network is represented by a digraph, the initial location of an infectious vehicle can be represented by a probability distribution of nodes. Then the worm propagation by V2V multi-hop communication can be described by a set of diffusion process on road network under IC model. For an infectious vehicle appearing at intersection s, each spanning path starting from intersection s is a possible path of worm propagation. Note that worm spreading in VANETs is mainly by broadcasting, so a sufficiently large number of spanning path starting from intersection s is required to approximate the broadcasting dissemination along diverse road sections.

We would like to restrict the worm propagation by deploying the MVSUs on some selected road sections. The size the area impacted by worm spreading will be restricted if the length of each single propagation path is limited. Assume the initial intersection s is the approximate location of the first infectious vehicular node. We define intersection r_i is reachable by the initial intersection s when there is at least one path $P(s, r_i)$ originating from intersection s to intersection r_i. For a limit number of MVSUs and a given road network, when the total number of r_i is minimized, the impact of a worm propagation is also minimized. Because of the uncertainty of the location of the initial infectious node, the deployment strategy should be capable of minimizing the reachable node number for an arbitrary intersection.

4.1.2 Vehicle Mobility

The objectives of MVSUs deployment are to sniff the infectious vehicles and to immunize the susceptible vehicles, so the vehicle mobility has to be incorporated. Instead of introducing other mobility models, we represent the vehicle by the same diffusion model used for describing the worm spreading by V2V communication.

The sniffing and disseminating blacklist are applied to individual a vehicle simultaneously and independently as long as the vehicle is passing by an MVSU. So the objective of MVSU deployment regarding the vehicle mobility can be rephrased as follows: We would like to deploy a limited number of MVSUs on the road network such that the probability of

a vehicle passes by an MVSU is maximized regardless the mobility pattern of the vehicle.

Assume the initial location of a vehicle is at intersection s. When a road graph is represented by a directed graph based on the prime approach, the potential routes of the vehicle are a set of spanning paths rooted at node s. A vehicle is passing by an MVSU refers to the corresponding spanning path rooted at node s is blocked by the MVSU. Then our optimization objective of MVSU deployment can be rephrased by blocking probability. For a limited number of available MVSUs, the aim is to find a set of road sections to deploy them such that the probability of blocking a potential infectious vehicle is maximized no matter where the vehicle appears and how it moves.

For a vehicle appearing node s, we assume the vehicle moves along the road and randomly selects one direction at each node. The vehicle could stop at an edge at any time. When a vehicle arrives at a node that has been visited already, it could be seen as another propagation process is about to start at the same node. The pass through probability p_i determines whether or not a vehicle would pass through or stop at the road section i. For each vehicle on a given road network, a probability distribution of p_i is used to describe its possible routes. The probability distribution of p_i governs the general displacement range of a vehicle so the distribution could be different from vehicle to vehicle. For a given road network, we would like to find a set of edges, by deploying MVSUs on them, the probability of successful "blocking" a vehicle on the road network is maximized regardless the pass through probability distribution of individual vehicle.

Similarly, the definition of reachable intersection could be employed to represent the goal of MVSU deployment strategy. For a vehicle appears at intersection s, a reachable intersection r_i is that at least one path $P(s, r_i)$ exists from intersection s to intersection r_i and there is not any MVSU on path $P(s, r_i)$. For a limited number of MVSUs, we would like the number of reachable nodes is minimized, which means, for a vehicle on the road network, the number of intersections that the vehicle could pass before it passes by an MVSU is minimized.

4.1.3 Problem Formulation

The definition of the reachable intersection is employed to show the conformities of the malicious behavior sniffing and blacklist immunization. Note that although both of the objectives can be represented by minimizing the number of reachable intersections, their probability distributions of the two diffusion processes are different.

To formulate our optimization objective, we define the free-path of node v on graph $G = (V, E)$ by $fp(v)$ such that $fp(v)$ starts at node v and there is no MVSUs on path

$fp(v)$. A vehicle moving along a free-path can neither be sniffed and nor receive an update of the blacklist. We define $G(v)$ is composed of all free-paths of v. With a limited number of special MVSUs, the node number of $G(v)$ is expected to be as small as possible for any $v \in V$.

For a vehicle appearing at the intersection corresponding to node v, the node number of $G(v)$ is the maximum number of reachable intersections that the vehicle can reach without being sniffed or receiving the blacklist update. We would like the node number of reachable intersections to be minimized no matter where the vehicle appears and how it moves. Ideally, it is possible to achieve the optimization objective by partitioning a graph into a set of subgraphs having similar or even equal number of intersections. We use the term "edge budget" to refer the number of available MVSUs. When edge budget is sufficiently large, there is always a set of edge cut can be found that parts a connected road network into some subgraphs in similar size. However, when the edge budget is less than the desired edge-cut number to do so, the road network cannot be reasonably parted or even cannot be entirely disconnected. In this case, the edges should be selected elaborately to achieve the optimization objective. Our study addresses the problem when the edge budget is less the cut number required to part the connected graph reasonably into a number of subgraphs.

4.2 Contamination Model on Graphs

In contamination control model, a network is represented by a directed graph. Each node has two states, "clean" and "contaminated". All the nodes are initialized to be clean. The contamination process is activated by a contamination source which is the node first changes its state spontaneously from clean to contaminated. The contamination source will try to impact its neighbors based on the propagation rule defined by the propagation model. Each newly impacted node will change its state from clean to contaminated, and the state changing cannot reverse. The state changing will enable a node to propagate the contamination based on the propagation model. This process repeats for each newly contaminated node then the contamination will be spread on the network. The contamination propagation stops when the stop conditions predefined in propagation model are achieved.

Finding a strategy to reduce the total contaminated nodes at the end of propagation process, known as the Minimum Contamination Problem (MCP), is one of the most popular objectives in this model. By blocking the propagation path, the propagation process would be altered or even be constrained at a particular part of the graph. The blocking strategies are various depends on scenarios, such as edge blocking strategies, node blocking strategies

and combination strategies. According to our scenario, the edge blocking strategy are interested.

To solve the proposed MVSU deployment problem, we introduce the MCP on graphs. The NP-hardness of the MCP has been discussed by Kimura et al. [69] and A. Li and L. Tang [74], so a heuristic method for the MVSU deployment problem is expected. We first build a graph from map data and then illustrate that the optimization objective of proposed MVSU deployment problem is the same as an MCP. After that, the solution to the MVSU deployment problem can be found by solving MCP on graphs.

4.2.1 Build Graphs from Map Data

Representing a road topology by a graph is preliminary for modeling the proposed MVSU deployment problem as the MCP. The prime approach is employed in our study. To keep the integrity, the details of representing the road network topology as a directed graph by the prime approach are explained here. A directed graph is built based on the intersections and road sections by representing the intersections and road sections as nodes and edges respectively. The direction of an edge is defined by the direction of the road section. When two neighboring intersections u and v are connected by a bidirectional road, both edge $\vec{e}_{uv}$ and edge $\vec{e}_{vu}$ are added on the corresponding directed graph $\vec{G}$. When intersections u and v are connected by a unidirectional road, for example from intersection u to v only, only edge $\vec{e}_{uv}$ is added to the directed graph $\vec{G}$.

As we discussed in Section 4.1, the topology used for worm spreading analysis should be able to represent the worm spreading by both V2V communication and physical carrying-spreading. The directed graph $\vec{G}$ is built based on the reachability of road intersections. However, the V2V multi-hop communication link is always bidirectional. When a V2V communication link exists between two neighboring intersections u and v, vehicles locating at intersection u and vehicles locating at intersection v are reachable to each other. Therefore, from a V2V communication aspect, for any edge $\vec{e}_{uv}$, there should be an edge $\vec{e}_{vu}$. As a result, the direction of each edge is ignored, and the directed graph $\vec{G}$ is converted to a simple undirected graph G.

The graph size influence the complexity of MCP, so the graph size is expected as small as possible. A road network built from map data can be simplified by removing redundant nodes and edges. For any graph G built from a road network, it could be simplified into a graph G' containing only nodes whose degree are greater than two. Think about a node with degree one which refers to a dead-end road in the real world. A vehicle originating at a node with degree one will be blocked by an MVSU as long as the vehicle originating from the

only neighbor of the degree-one node will be blocked. So the node with degree one can be merged into its only neighbor node. Similarly, for a node with degree two, a vehicle starting to move from a degree-two node will always be blocked as long as the vehicle originating from either neighbor of the degree-two node will be blocked. As a result, any path without branch is a sub-structure that can be simplified as an edge. The path is replaced by an edge originating from the starting node and pointing to the ending node of the path. The pass through probability equivalence of replacing a path $P = \{v_0, v_1, v_2, ..., v_i\}$ by an edge $\overrightarrow{e}_{v_0 v_i}$ can be derived as follows: a path P and probabilities $p_{\overrightarrow{e}_{v_0 v_1}}, p_{\overrightarrow{e}_{v_1 v_2}}, p_{\overrightarrow{e}_{v_2 v_3}}, ..., p_{\overrightarrow{e}_{v_{i-1} v_i}}$ of each edge on path P, the probability of $\overrightarrow{e}_{v_0 v_i}$ is $p_{\overrightarrow{e}_{v_0 v_i}}$. The mathematical expression of $p_{\overrightarrow{e}_{v_0 v_i}}$ can be found in Equation 4.1.

$$p_{\overrightarrow{e}_{v_0 v_i}} = \prod_{n=1}^{i} p_{\overrightarrow{e}_{v_{n-1} v_n}} \tag{4.1}$$

After this simplification procedure, the simplified graph G' contains only nodes with degrees greater than two, and it still keeps the reachability feature of the original road network. Figure 4.1b shows an example of the simplified graph built from road map data of Ottawa downtown area (shown in Figure 4.1a).

4.2.2 Minimum Contamination Problem

We model the proposed MVSU deployment problem as minimum contamination problem (MCP) by representing the route of worm spreading and vehicle movement as a contamination propagation process. Similar with contamination prorogation, the possible route of the worm spreading and the vehicle movement on road networks is stochastic. We assume the probability of an infectious vehicle initially appears at a node v_i follows a probability distribution $p(v_i)$. The initial location of an infectious vehicle can be represented by a contamination source node which is the first node turning to "contaminated". The vehicle moving on a road network is equivalent to a contamination propagation process on a directed graph $\overrightarrow{G}$ built from the road network. For example, an infectious vehicle appearing at intersection v_0 on a road network randomly selects a direction to go at the intersection and it repeatedly random selects a direction to go at each intersection. After i intersections, it tries to pass through a road section $\overrightarrow{e}_{v_i v_{i+1}}$. There is a MVSU on the road section $\overrightarrow{e}_{v_i v_{i+1}}$, then the worm is sniffed. A path $\{v_0, v_1, v_2, ..., v_i\}$ can be found on the corresponding directed graph, and its step length is i. The path $\{v_0, v_1, v_2, ..., v_i\}$ of the vehicle is equivalent with a contamination propagation process along the same path starting at node v_0 and being terminated at edge $\overrightarrow{e}_{v_i v_{i+1}}$.

(a) Original map data [53]

(b) Graph built from map data [53]

Figure 4.1: Original Map and the Graph after Simplification

In Section 4.1, the worm spreading by both V2V multi-hop communication and physical carrying could be represented by IC model. Therefore, our optimization objectives can be rephrased by MCP under IC model. For a predefined number of available MVUSs K_{eb} and a directed graph $\vec{G}$, at a time t, node v_0 turns its state from "clean" to "contaminated" and selects one of its neighbors to send the contamination message. There is a probability p_{e_i} on each edge e_i determining whether or not the contamination message could successfully pass through the edge. If the contamination message successfully passed through, the selected neighbor will turn to "contaminated" and choose one neighbor to send contamination message. Each newly contaminated node repeats these procedures, until, at time $t+n+1$, a failure occurs when a contamination message is passing through an edge because of probability p_{e_i}. By the end of a contamination propagation process, the contaminated node number by the source node v_0 is expected to be minimized by blocking the solution edge set $\vec{E}_s$, where $\vec{E}_s \in \vec{E}$ and $|\vec{E}_s| = K_{eb}$.

To further quantify the optimization objective, the definitions of contamination degree in [71] is introduced. For a node v on digraph $\vec{G} = (V, \vec{E})$, the contamination degree of v, denoted by $C(v)$, is the number of contaminated nodes in the end of a contamination process originating from node v. Average contamination degree of graph $\vec{G} = (V, \vec{E})$, denoted by $C_{ave}(\vec{G})$, is the mean value of contamination degrees of all v_i, $v_i \in V$.

$$C_{aver}(G) = \frac{1}{|V|} \sum_{i=1}^{n} C(v_i), \{v_i \in V \quad and \quad i \in [1, n]\} \tag{4.2}$$

Worst contamination degree of graph $\vec{G} = (V, \vec{E})$, denoted by $C_{worst}(v)$, is the maximum value of contamination degrees of all nodes $\{v_i,\, v \in V\}$.

$$C_{worst}(G) = max\{C(v_i), v_i \in V \quad and \quad i \in [1, n]\}. \tag{4.3}$$

However, neither of these two definitions incorporate the probability distribution of the initial location of the source node. In our model, the probability distribution p_i is required for describing the initial location of the vehicle. So, we generalize the MCP problem by introducing expected contamination degree. The probability of node v_i to be the contamination source node follows a distribution $p(v_i)$. Then the expected contamination degree can be formulated as Equation 4.4.

$$C_{ex}(G) = \sum_{i=1}^{n} p(v_i) C(v_i), \{v_i \in V \quad and \quad i \in [1, n]\} \tag{4.4}$$

It is notable that the average contamination degree is the expected value of contamination degree when each node has equal probability being selected as the contamination source node. So when $p(v_i)$ follows the uniform distribution, the value of expected contamination degree is the same as the average contamination degree.

Our optimization objective is to minimize the expected contamination degree when the probability of infectious vehicle follows a given probability distribution. The expected contamination degree is reduced by removing a set of edges. For a given directed graph $\vec{G}$ and an edge budget K_{eb}, the minimum expected contamination degree edge blocking strategy is an edge set $\{\vec{E}_{ex}, |\vec{E}_{ex}| = K_{eb}\}$ such that, by removing $\vec{E}_{ex}$, the expected contamination degree of graph $\vec{G}\backslash\vec{E}_{ex}$ is minimum for all $\vec{G}\backslash\vec{E'}_s$, $\{|\vec{E}_{ex}| = K_{eb}, |\vec{E'}_s| = K_{eb}\}$. Note that the simplified graph G built by the procedure in Section 4.2.1 is a undirected graph. The graph G is converted into a directed graph $\vec{G}$ such that each edge e_{uv} is replaced by an edge pair $\vec{e}_{uv}$ and $\vec{e}_{vu}$ so that the minimum contamination model can be applied.

Chapter 5

Proposed Methods

Based on the existing studies, the most challenge of solving MCP on large-scale networks, such as road networks and computer networks, is the exponential increasing complexity for the contamination degree estimation. The greedy method proposed in [71] somehow solves this scalability issue by bond percolation approximation, so it is worth to explore the possibility to apply this method to road networks. In this chapter, we first transplant the existing method for social networks to road networks and then develop two methods for solving MCP on VANETs environment.

5.1 Original Greedy Method (OGM)

The greedy method proposed in [71] is practical for a large-scale network, so it is possible to apply it to road networks. This method is referred as Original Greedy Method (OGM) in the following discussion. The general idea of the Original Greedy Method proposed in [71] is to find the edges having the most influence to the contamination degree, which can be summarized as follows: Let $G(e_i)$ denotes the graph G excluding edge e_i. For a given directed graph $\overrightarrow{G}$, only one edge $\{\overrightarrow{e}_i, \overrightarrow{e}_i \in \overrightarrow{E}\}$ is blocked at one time, and then the average and the worst contamination degree of $\overrightarrow{G}(\overrightarrow{e}_i)$ are calculated to assess the influence of blocking edge $\overrightarrow{e}_i$. After all the edges are evaluated, the edges can be sorted by their contamination degrees in ascending order. The top K_{eb} edges with the minimum value of the average contamination degree and the top K_{eb} edges with the minimum value of the worst contamination degree could be selected, where K_{eb} is the edge budget.

By using the same procedure as the greedy method proposed in [71], the edge sets to the minimum expected contamination degree and the minimum worst contamination degree can found. Let $C_{ex}(\overrightarrow{G}(\overrightarrow{e}_i))$ and $C_{wst}(\overrightarrow{G}(\overrightarrow{e}_i))$ denote the expected and worst contamination

degree of graph $\vec{G}(\vec{e}_i)$ respectively. Let $\vec{E}_{ex}$ and $\vec{E}_{wst}$ be the solution edge sets to the minimum expected contamination degree problem and the minimum worst contamination degree problem, accordingly. For every edge $\vec{e}_i$, $C_{ex}(\vec{G}(\vec{e}_i))$ and $C_{wst}(\vec{G}(\vec{e}_i))$ are evaluated. Then the edge set $\vec{E}_{ex}$ and $\vec{E}_{wst}$ can be found by sorting the edge $\vec{e}_i$ according to the value of $C_{ex}(\vec{G}(\vec{e}_i))$ and $C_{wst}(\vec{G}(\vec{e}_i))$, respectively. For a given K_{eb}, the top K_{eb} edges in $\vec{E}_{ex}$ are the solution to the expected contamination minimization problem and top K_{eb} edges in $\vec{E}_{wst}$ are the solution to the worst contamination minimization problem.

The challenge of implementing this greedy method is that the amount of calculation will explosively increase along with the increase of the size of input graph. Because the contamination propagation is a stochastic process, a sufficient large iteration number is required to estimate $C(v_i)$ for each node $v_i \in V$. Assume M is a sufficient large integer, $C(v_i)$ can be estimated by Equation 5.1 [71]. Then the expected contamination degree after blocking edge $\vec{e}_i$ can be found by Equation 5.2. Similarly, the worst contamination degree $C_{wst}(\vec{G}(\vec{e}_i))$ can be found by the maximum value of $C(v_i)$ (shown as Equation 5.3). In order to sort all the edges, each edge $\vec{e}_i \in \vec{E}$ has to be blocked alone to calculate $C_{ex}(\vec{G}(\vec{e}_i))$ and $C_{wst}(\vec{G}(\vec{e}_i))$.

$$C(v_i) = \frac{1}{M}\sum_{m=1}^{M} C^m(v_i) \tag{5.1}$$

$$C_{ex}(G(\vec{e_i})) = \sum_{i=1}^{n} p(v_i)C(v_i) \tag{5.2}$$

$$C_{wst}(G(\vec{e_i})) = max\{C(v_i), v_i \in V\} \tag{5.3}$$

According to the discussion in [70], the total complexity of the greedy method is $|\vec{E}| \times |V| \times M \times O(C(v_i))$, where $|\vec{E}|$ and $|V|$ are the edge and node numbers of graph $\vec{G}$, M is the iteration number to estimate $C(v_i)$, and $O(C(v_i))$ is the complexity of calculating $C(v_i)$. When the graph is large, the amount of calculation becomes impractical. Therefore, bond percolation approximation was proposed in [70] to reduce the amount of calculation for estimating the contamination degree of node v_i.

5.1.1 Bond Percolation and Percolation Threshold

Percolation theory was first proposed by Broadbent and Hammersley in 1956 [42]. It was designed for describing the stochastic movement and the expansion of the fluid in a porous

medium. When the number of blocked holes reaches a threshold, the fluid cannot pass the porous medium. This property has been widely used in the research of disordered system and stochastic process. Based on this physical phenomenon, two percolation models, bond percolation and node percolation, are developed for different research interests. The equivalence of IC model and bond percolation model has been discussed in [72], [79] and [44].

The bond percolation of a graph G is a process that the appearance (also named as "occupation") of each edge e_i depends on the probability p_{e_i}, where p_{e_i} is called percolation probability of edge e_i. Percolation probability could be a uniform value for all edges or a predefined probability distribution for each edge e_i. A uniform percolation probability means each edge e_i has identical value of p_{e_i}. Percolation probability p_{e_i} could also be a function depends on other parameters such as degrees of the nodes attaching to the edge. The bond percolation process for the directed graph can be found in Procedure 1, where p_p is the given percolation probability, $\{p, 0 \leqslant p \leqslant 1\}$ is a randomly generated number, $\vec{G}_p$ is the output graph by bond percolation process.

Procedure 1 Bond Percolation (Directed Graphs)

procedure BOND PERCOLATION($\vec{G} = (V, \vec{E}), p_p$)
  Initialize: $\vec{G}_p = (V, \vec{E}_p), \vec{E}_p \leftarrow \varnothing$
  for all $\vec{e}_i \in \vec{E}$ **do**
    if $p \leqslant p_p$ **then** ▷ p is a random number, $0 \leqslant p \leqslant 1$
      $\vec{E}_p \leftarrow \vec{E}_p \cup \{\vec{e}_i\}$
    end if
  end for
  return $\vec{G}_p$
end procedure

The percolation process at the probability of percolation threshold is most interested. The percolation threshold is found when a graph reaches a critical state of the existence of a giant cluster. In perfect condition, when percolation process is carried out above the percolation threshold, a giant cluster will always exist. When percolation process is performed under the percolation threshold, the giant cluster will always vanish. The number of absent edges at critical state can be seen as a loose upper bound of the number of edges required to be removed that guarantees no giant cluster exists. In other words, for a given graph $\vec{G}$, there will be no giant cluster if randomly removed $|\vec{E}|(1 - p_{th})$ edges from the graph $\vec{G}$, where p_{th} is the percolation threshold of $\vec{G}$. Therefore, the critical state of the existence of giant cluster is also the critical state of that whether or not we

can always find an edge cut to constrain the contamination within a subgraph of $\vec{G}$. So the percolation threshold of a graph $\vec{G}$ is used as the uniform percolation probability to estimate the contamination degree of node v_i.

The bond percolation threshold can be found when the size of second largest cluster is maximum. The strongly connected compone (SCC) is used as the cluster to estimate the bond percolation threshold. The bond percolation probability increases from 0.01 to 0.99 with the step length of 0.01. The the size of the second largest SCC is calculated and averaged by running percolation K_p times at each percolation probability p_p. The percolation threshold estimation is implemented as shown in Procedure 2.

Procedure 2 Bond Percolation Threshold Estimation (Directed Graphs)

procedure BOND PERCOLATION THRESHOLD ESTIMATION($\vec{G} = (V, \vec{E}), K_p$)
  Initialize: $\vec{G_p} \leftarrow \varnothing, \overline{SLSCC} \leftarrow 0, SLSCC_{max} \leftarrow 0, p_{th} \leftarrow 0, p_p \leftarrow 0, sl_p \leftarrow 0.01, K \leftarrow 0$
  while $p_p \leqslant 1$ **do**
    $SLC_{total} \leftarrow 0$
    while $K < K_P$ **do**
      $\vec{G_p} \leftarrow$ PERCOLATION($\vec{G} = (V, \vec{E}), p_p$)
      $SLC_{total} \leftarrow SLSCC_{total} + SLSCC$ ▷ SLSCC is the node number of the second largest strongly connected component.
      $K \leftarrow K + 1$
    end while
    $\overline{SLSCC} = SLSCC_{total}/K_p$
    if $SLC_{max} < \overline{SLSCC}$ **then**
      $SLC_{max} \leftarrow \overline{SLSCC}$
      $p_{th} \leftarrow p_p$
    end if
    $p_p \leftarrow p_p + sl_p$
  end while
  return p_{th}
end procedure

5.1.2 Bond Percolation Approximation

As a matter of fact, bond percolation imitates the stochastic process of contamination propagation parallel starting from all nodes on the graph, so the amount of calculation

can be reduced remarkably. For a directed graph $\vec{G}$, each single bond percolation iteration produces a subgraph $\vec{G_p} = (V, \vec{E_p})$ of original directed graph, where $\vec{E_p} \subset \vec{E}$. The reachable nodes from node v_i are equivalent with the nodes contaminated by the contamination process starting from node v_i under IC model. By calculating the reachable node number of each node v_i on graph $\vec{G_p}$, the contamination degree of every node on graph $\vec{G_p}$ can be found. The contamination degree calculated based on $\vec{G_p}$ is an approximate result of blocking edges that are absent on $\vec{G_p}$. A single bond percolation result at probability p_p can be used for evaluating the impact of roughly removing $|\vec{E}|(1-p_p)$ edges on $\vec{G}$ so that a considerable calculation reduction can be achieved by applying bond percolation approximation.

With the approximation by bond percolation, it is feasible to apply the greedy method on large scale networks such as road networks. Recall the greedy method, for a given edge budget K_{eb}, the edge set to solve expected contamination degree minimization problem, $\{\vec{E}_{ex}, |\vec{E}_{ex}| = K_{eb}\}$, can be found by the following procedure: First, for each edge $\vec{e}_i$ in graph $\vec{G}$, $C(\vec{G}(\vec{e}_i))$ is calculated. Then, the $C(\vec{G}(\vec{e}_i))$ is sorted in an ascending order. The top K_{eb} edges with most minimum $C(\vec{G}(\vec{e}_i))$ are added to the selected edge set $\vec{E}_{ex}$.

By bond percolation approximation, the calculation reduction techniques used in [71] can be employed here to further reduce the amount of calculation for calculating $C(\vec{G}(\vec{e}_i))$ by Equation 4.4. The techniques are summarized as following: Instead of running percolation process M iterations for each $\vec{G}(\vec{e}_i)$, percolation process is carried out M iterations on $\vec{G}$, then a set of graphs, $\left\{\vec{G_p^1}, \vec{G_p^2}, \vec{G_p^3}, ..., \vec{G_p^m}\right\}$, will be generated. Then a subset S_{e_i} of $\left\{\vec{G_p^1}, \vec{G_p^2}, \vec{G_p^3}, ..., \vec{G_p^m}\right\}$ could be found, such that, for any $\vec{G_p^i} \in S_{e_i}$, edge $\vec{e}_i$ is absent in $\vec{G_p^i}$. When M is sufficient large, $|S_{e_i}|$ is also sufficient large. Then the expected contamination degree of blocking edge $\vec{e}_i$ can be calculated by all the graphs $\vec{G_p^i} \in S_{e_i}$, where S_{e_i} is the subset of $\left\{\vec{G_p^1}, \vec{G_p^2}, \vec{G_p^3}, ..., \vec{G_p^m}\right\}$ such that $\vec{e}_i \notin \vec{G_p^i}$ for every $\vec{G_p^i} \in S_{e_i}$ [70].

Equation 5.4 calculates the expected contamination degree of $\vec{G}(\vec{e}_i)$. The expected contamination degree of each bond percolation result $\vec{G_p^i}, i \in [1, m]$ can be calculated by Equation 5.5 which is the standard way to calculate expected contamination degree. However, according to [71], the nodes in the same strongly connected component (SCC) have the same number of reachable nodes. So SCC condensation can be used to reduce the amount of calculation. The graph $G_p^i, i \in [1, m]$ is condensed into a number of SCCs, $\{SCC_1, SCC_2, SCC_3, ..., SCC_J\}$, the node number of SCC_i is denoted by $|SCC_i|$. The expected contamination degree can be calculated by Equation 5.6, where p_i is the proba-

bility of node v_i to be source node and $v_i \in SCC_j$.

$$C_{ex}(G(e_i)) = \frac{1}{|S_{e_i}|} \sum_{m=1}^{|S_{e_i}|} C_{ex}(G_p^m) \tag{5.4}$$

$$C_{ex}(G_p^m) = \sum_{i=1}^{n} p(v_i)C(v_i), \{v_i \in G_p^m\} \tag{5.5}$$

$$C_{ex}(G_p^m) = \sum_{j=1}^{J} \sum_{i=1}^{|SCC_j|} p(v_i)C(SCC_j) \tag{5.6}$$

5.1.3 Algorithm Description of OGM

The full algorithm of the Original Greedy Method is shown in algorithm 1. The inputs of this algorithm are a directed graph $\vec{G}$ built from road data, a positive integer K_{eb}, and a probability distribution $P(v)$ presenting the probability of node v to be the initial location of the infectious vehicle.The output is an edge set to deploy the K_{eb} available MVSUs to inhibit the worm spreading on the road network.

The algorithm starts by variables initialization. The expected contamination degree and the worst contamination degree of blocking edge $\vec{e}_i$ denoted by $EC_{\vec{e}_i}$ and $WC_{\vec{e}_i}$ are initialized as 0, and the solution edge list to the minimum expected contamination degree and the minimum worst contamination degree are denoted by $\vec{E}_{ex}$ and $\vec{E}_{wst}$ respectively. Both $\vec{E}_{ex}$ and $\vec{E}_{wst}$ are initialized as an empty set. The bond percolation approximation is carried out under percolation threshold, so the first step of this procedure is to estimate the value of percolation threshold of the input graph $\vec{G}$. The Procedure 2 is applied to the input directed graph for the bond percolation threshold estimation. The output of Procedure 2 is the bond percolation threshold p_{th}. Based on the percolation threshold p_{th}, the M iterations of the bond percolation process are implemented. In each iteration, a directed graph $\vec{G}_i$ is built by running the bond percolation module shown in Procedure 1. The contamination degree of each node v on $\vec{G}_i$ (denote by $C(v)$) is calculated by Equation 5.6. Then its expected contamination degree $C_{ex}(v)$ can be found by taking the initial location probability distribution $P(v)$ into account. Meanwhile, the worst contamination degree is compared with $C(v)$ to ensure the largest contamination degree value is stored in C_{wst}. After that, the expected contamination degree of every node v in graph $\vec{G}_i$ is accumulated by C_{ex}. The expected contamination degree and the worst contamination degree of $\vec{G}_i$ is a result of removing all the edges in the set $\{\vec{G} \setminus \vec{G}_i\}$. So the C_{ex} and C_{wst} are assigned to the edges absent on graph $\vec{G}_i$. After M bond percolation iterations, the average value

Algorithm 1 Original Greedy Method

Input: $\vec{G} = (V, \vec{E}), M, K_p$

Output: $\vec{E}_{ex}, \vec{E}_{wst}$

Initialize: $m \leftarrow 0, C_{ex} \leftarrow 0, C_{wst} \leftarrow 0, \leftarrow 0, \vec{E}_{ex} \leftarrow \varnothing, \vec{E}_{wst} \leftarrow \varnothing$
for all $\vec{e}_i \in \vec{G}$ **do**
    $EC_{\vec{e}_i} \leftarrow 0$
    $WC_{\vec{e}_i} \leftarrow 0$
    $counter_{\vec{e}_i} \leftarrow 0$
end for
$p_{th} \leftarrow$ Bond Percolation Threshold Estimation$(\vec{G} = (V, \vec{E}), K_p)$
while $m < M$ **do**
    $\vec{G^m}_p \leftarrow$ Percolation$(\vec{G} = (V, \vec{E}), p_{th})$
    for all $v_i \in \vec{G^m}_p$ **do**
        $C_{ex} \leftarrow C_{ex} + p(v_i) * C(v_i)$
        if $C_{wst} < C_{v_i}$ **then**
            $C_{wst} \leftarrow C(v_i)$
        end if
    end for
    for all $\vec{e}_i \in \vec{G}$ and $\vec{e}_i \notin \vec{G^m}_p$ **do**
        $EC_{\vec{e}_i} \leftarrow EC_{\vec{e}_i} + C_{ex}$
        $WC_{\vec{e}_i} \leftarrow WC_{\vec{e}_i} + C_{wst}$
        $counter_{\vec{e}_i} \leftarrow counter_{\vec{e}_i} + 1$
    end for
    $m \leftarrow m + 1$
end while
for all $\vec{e}_i \in \vec{G}$ **do**
    $\overline{EC}_{\vec{e}_i} \leftarrow (EC_{\vec{e}_i}/counter_{\vec{e}_i})$
    $\overline{WC}_{\vec{e}_i} \leftarrow (WC_{\vec{e}_i}/counter_{\vec{e}_i})$
end for
$\vec{E}_{ex} \leftarrow$ sorting: $\vec{E}$ by $\overline{EC}_{\vec{e}_i}$
$\vec{E}_{wst} \leftarrow$ sorting: $\vec{E}$ by $\overline{WC}_{\vec{e}_i}$

of expected contamination degree and worst contamination degree of blocking each edge e can be found. To achieve this, $counter_e$ is used to count the number of values being accumulated. The average values of expected contamination degree and worst contamination degree of blocking edge e are denoted by $\overline{EC}_{\vec{e}_i}$, and $\overline{WC}_{\vec{e}_i}$ respectively. By sorting $\overline{EC}_{\vec{e}_i}$, and $\overline{WC}_{\vec{e}_i}$ and in ascending order, the edge set $\vec{E}_{ex}$ and $\vec{E}_{wst}$ can be found. For a given K_{eb}, the top K_{eb} edges of $\vec{E}_{ex}$ are the solution to minimum expected contamination problem and the top K_{eb} edges of $\vec{E}_{wst}$ are the solution to minimum worst contamination problem. The top K_{eb} edges from $\vec{E}_{ex}$ are the edge set used as the solution to the MVSU deployment problem on the road network.

5.2 Grid-shrinking Greedy Method (GGM)

The greedy method is applicable for minimum contamination problem on both road networks and social networks, which inspires us to develop methods for VANETs and road networks based on it. The second proposed method is a greedy method improved by taking typical road patterns into account. It is referred as Grid-shrinking Greedy Method (GGM) in the following discussion.

There are four modules in our second proposed method: grid detection module, grid shrinking module, greedy edge ranking module and edge mapping module. Similar to method one, the directed graph $\vec{G}$ built from map data is used as the input for proposed method two. Grid detection and grid shrinking modules detect and shrink the grid patterns on the road network into pseudo-vertices. Greedy edge ranking module is the same as the greedy method in Algorithm 1. It finds the solution edge sets on the graph after grid shrinking. In the end, the edge mapping module finds the edge set in the original directed graph based on the edges ranking result of the graph after grid shrinking.

5.2.1 Grid Detection and Shrinking

Grid (square lattice) is a typical road pattern in road networks [65]. The bond percolation threshold of a undirected graph composed by infinite square lattice is known as 0.5. A 100×100 grid is built to illustrate the bond percolation threshold of a directed graph and a undirected graph of the same grid structure. The bond percolation threshold curves of the grid drawn by Procedure 2 and Procedure 7 are shown in Figure 5.1.

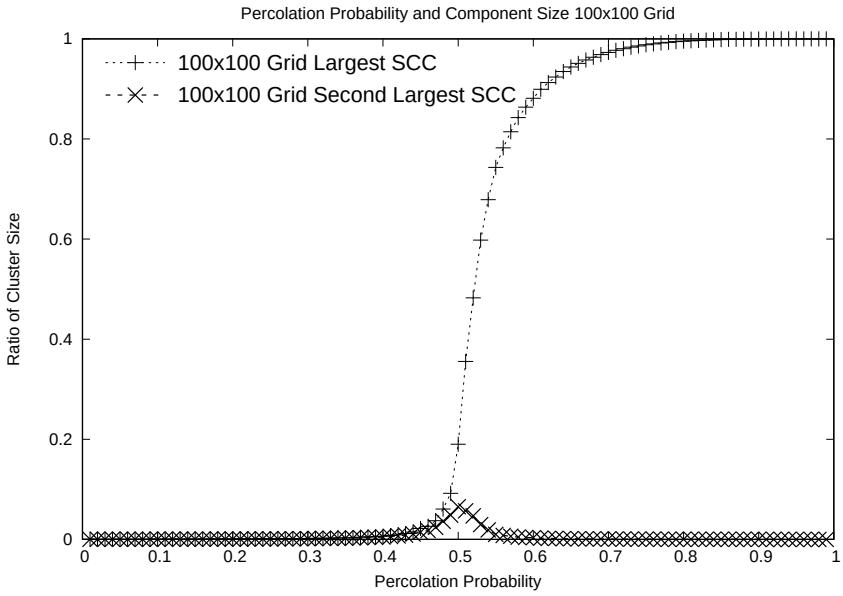
The bond percolation threshold of a directed graph is found when the size of second largest strongly connected component (SCC) is maximized. The bond percolation threshold

of the undirected graph is found when the size of second largest connected component (CC) reaches the maximum. The SCC and CC are clusters on a graph, so the Ratio of Cluster Size defined by the ratio of cluster node number to the total node number of the graph is introduced to measure the size of SCC and CC. For a given input graph, the curves of largest cluster and second largest cluster with different percolation probability are drawn to observe their size variation. According to the results in Figure 5.1, the size of second largest cluster reach maximum when percolation probability is 0.5, so the bond percolation thresholds of both the directed and undirected grids are 0.5.

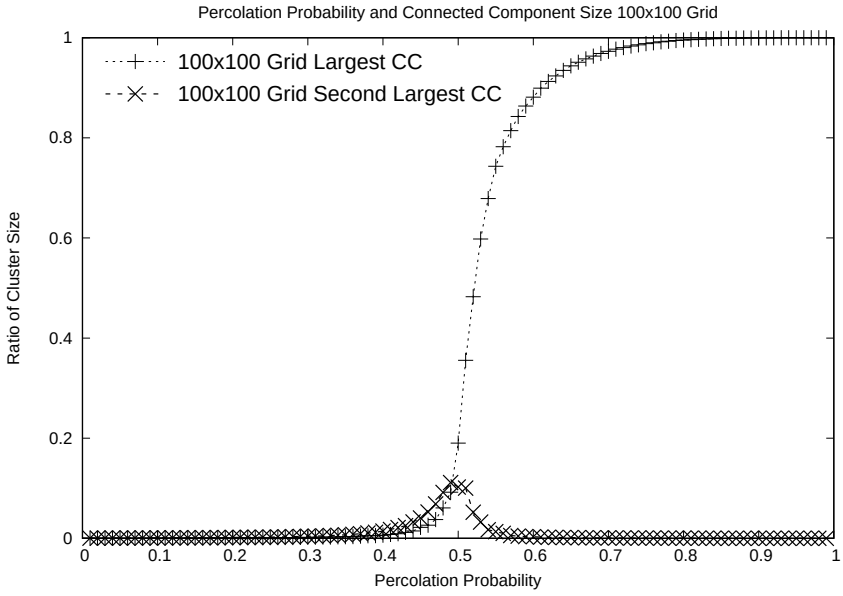
Based on our simulation results, it is notable that the bond percolation threshold values of all six graphs built from road networks are greater than the bond percolation threshold of the grid structure (see Table 6.2). According to the percolation theory, the number of connected nodes will sharply increase when a percolation is carried out with a percolation probability that is greater than the percolation threshold. So when the percolation is performed on a graph containing grid subgraphs at a probability that is greater than 0.5, the reachable node number on grid subgraphs are dramatically increased. From a contamination propagation aspect, when any node in the grid structure is contaminated, there is always a path propagating the contamination to a node outside the grid structure. In this case, the grid structure can be seen as one pseudo vertex that has the capability to propagate the contamination to any node connecting to this pseudo vertex. By this approximation, the node number and edge number of the input graph could be significantly reduced which further reduces the amount of calculation in bond percolation process. To apply this approximation, the grid structures in road networks have to be detected first.

Grid patterns in road networks are identified by the grid detection module. The process of grid detection is shown as Procedure 3. We are looking for the grid-like structures instead of strictly grid structures on road networks due to the concerns of the diversities of road networks. For a given undirected graph built from map data, the nodes with degree four are detected first. Let G_{grid} be an empty graph that is used to store the identified grid structures. First, the frame of grid-like structures is built by adding all nodes with degree four and their neighboring nodes on road network into sub-graph G_{grid}. And then, traverse all the nodes in G but not in G_{grid} to search for the common neighbor of two nodes of G_{grid} and add the common neighbors to G_{grid}. The common neighbor nodes are recursively added to G_{grid}, till no new common neighbor nodes could be found. In the end, the pruning process repeatedly removes the nodes with degree one in G_{grid} till all the degree-one nodes are removed. Then each connected sub-graph in G_{grid} is a grid-like structure.

The grid shrinking process is shown as Procedure 4. The edges connecting to a grid-like structure are handled elaborately by this module because the edge set from the original graph is expected as the final solution set but the auxiliary edges added on the graph after



(a) 100x100 Grid (Directed)



(b) 100x100 Grid (undirected)

Figure 5.1: Bond Percolation Threshold of the Grid

Procedure 3 Grid Detection

1: **procedure** GRID DETECTION($G = (V, E)$) ▷ G is a undirected graph
2: **Initialize:** $G_{grid} \leftarrow \varnothing, E_{tba} \leftarrow \varnothing, check_1 \leftarrow 1, check_2 \leftarrow 1$
3: **for all** $v \in G$ **do**
4: **if** $|N(v)| = 4$ **then** ▷ node v has 4 neighbors
5: **for all** $u \in N(v)$ **do**
6: $G_{grid} \leftarrow G_{grid} \cup \{e_{uv}\}$
7: **end for**
8: **end if**
9: **end for**
10: **while** $check_1 > 0$ **do**
11: $check_1 \leftarrow 0$
12: **for all** node $v \in G_{grid}$ **do**
13: **if** $\{|N(v)| = 4\} \wedge \{N(u_1) \cap N(u_2) = \{v, w\}, (u_1, u_2 \in N(v))\}$ **then** ▷ node u_1 and u_2 are two neighbors of node v and they have two common neighbors v and w
14: $E_{tba} \leftarrow E_{tba} \cup \{e_{wu_1}, e_{wu_2}\}$
15: $check_1 \leftarrow check_1 + 1$
16: **end if**
17: **end for**
18: $G_{grid} \leftarrow G_{grid} \cup E_{tba}$
19: $E_{tba} \leftarrow \varnothing$
20: **end while**
21: **while** $check_2 > 0$ **do**
22: $check_2 \leftarrow 0$
23: **for all** $\{v \in G_{grid}\} \wedge \{|N(v)| = 1\}$ **do** ▷ node v has only one neighbor
24: $G_{grid} \leftarrow G_{grid} \setminus \{v\}$
25: $check_2 \leftarrow check_2 + 1$
26: **end for**
27: **end while**
28: **return** G_{grid}
29: **end procedure**

grid shrinking. Edge set E^i_{n2v} is defined by node-to-vertex edge set connecting to G^i_{grid}. Edge set E^{ij}_{v2v} is defined by vertex-to-vertex edge set connecting grid G^i_{grid} and grid G^j_{grid}. Graph G_{gs} is the grid shrinking result which is initialized as an empty graph. E^i_{n2v} and E^{ij}_{v2v} are also initialized to be empty. For every edge e_{uv} on graph G, node u and v are examined respectively by whether or not the node belongs to a pseudo-vertex. The edges linking two pseudo-vertices are stored in the edge set E^{ij}_{v2v}, where i, j are the indexes of the pseudo-vertices. The edges connecting a node and a pseudo-vertex are stored in the edge set E^i_{n2v}, where i is the index of the pseudo-vertex. The edges connecting two normal nodes are added to the graph directly G_{gs}.

Procedure 4 Grid Shrinking

procedure GRID SHRINKING($G = (V, E), G_{grid}$)) ▷ G is a undirected graph
  Initialize: $G_{gs} \leftarrow \varnothing$
  for all $e_{uv} \in G$ **do** ▷ check each edge in G
    if $u \in G^i_{grid}$ **then**
      if $\{v \in G^j_{grid}\} \wedge \{(i \neq j)\}$ **then** ▷ nodes u,v belong to different grids
        $E^{ij}_{v2v} \leftarrow E^{ij}_{v2v} \cup e_{uv}$
        $G_{gs} \leftarrow G_{gs} \cup e_{ij}$
      else if $u \in G^j_{grid}, (i = j)$ **then**
        continue ▷ both nodes belong to one grid, skip
      else ▷ only one node belongs to a grid
        $E^i_{n2v} \leftarrow E^i_{n2v} \cup e_{iv}$
        $G_{gs} \leftarrow G_{gs} \cup e_{iv}$
      end if
    else if $v \in G^i_{grid}$ **then** ▷ node v belongs to grid i
      $E^i_{n2v} \leftarrow E^i_{n2v} \cup e_{ui}$
      $G_{gs} \leftarrow G_{gs} \cup e_{ui}$
    else
      $G_{gs} \leftarrow G_{gs} \cup e_{uv}$ ▷ nodes u,v are not in any grid
    end if
  end for
  return$G_{gs}, \{E^1_{n2v}, E^2_{n2v}, E^3_{n2v}, ..., E^i_{n2v}\}, \{E^{12}_{v2v}, E^{13}_{v2v}, E^{14}_{v2v}, ..., E^{ij}_{v2v}\}$
end procedure

Figure 6.2 shows the grid detection results of the six maps used for simulation. It can be found that the size of detected grid structures heavily depends on the road topologies. In Figure 6.2b, the detected grid structures are small because the road topology is less regular than others. In Figure 6.2a, 6.2c and 6.2d, there are many regular blocks in the

map, so the detected grid structures took a significant ratio of total road sections. The graphs built from map 5 and 6 are composed of several large grid structures, so, as shown in Figures 6.2e and 6.2f, most of the road sections belong to grid structures.

The grid shrinking results are shown in Figure 6.3. The ratio of grid structures significantly influences the characteristics of the graph after grid shrinking. By comparing the graph shown in Figure 6.2b and 6.3b, it can be found that the grid shrinking has little influence on the general topology, because the size of grid structures found in graph 6.2b is much less than the size of the original graph. In the graphs shown in Figure 6.3a, 6.3c, and 6.3d, the general road topology are kept the same, but some vertices with a high degree number are created. In the graphs shown in Figure 6.3e and 6.3f, most of the rod sections are shrunk into pseudo-vertices and only a small part of the road sections are kept. Therefore, from the grid shrinking results, we can see that when the size of grid structure increase, the grid shrinking procedure may considerably change the graph feature.

After grid detection, the contamination degree of blocking the edges in the grids are not necessarily to be examined. When the edge budget is much less than the total edge number of graph G, there is not sufficient edge budget to fairly disconnect the graph G into several small subgraphs. The edges contributing to general connectivity is more influential than the edges in local structures from the overall contamination degree reduction aspect.

5.2.2 Edge Mapping

The edge set E_{exgs} (solution edge set to minimum expected contamination problem on the graph after shrinking) found by greedy edge ranking module is a set of undirected edges in G_{gs}. However, the input graph is a directed graph instead of a undirected graph after grid-shrinking. So, the edge mapping module is required to find the corresponding directed edges of the original directed graph $\vec{G}$ based on the edge set E_{exgs}.

There are three situations when mapping an edge $e_{uv} \in E_{exgs}$ back to the directed graph $\vec{G}$ built from the road network:

1. Both $\vec{e}_{uv}$ and $\vec{e}_{vu}$ exist in $\vec{G}$.
2. Only $\vec{e}_{uv}$ exists in $\vec{G}$.
3. Neither $\vec{e}_{uv}$ nor $\vec{e}_{vu}$ exist in $\vec{G}$.

The edge mapping module judges the situation of each edge $e_{uv} \in E_{exgs}$ and handles the situation correspondingly. The details are explained as below.

For the edge e_{uv} with sequence number k, the existence of $\overrightarrow{e}_{uv}$ and $\overrightarrow{e}_{vu}$ are judged first. When both $\overrightarrow{e}_{uv}$ and $\overrightarrow{e}_{vu}$ exist in $\overrightarrow{G}$, e_{uv} can be replaced by $\overrightarrow{e}_{uv}$ and $\overrightarrow{e}_{vu}$. k and $k+1$ are assigned as sequence number for $\overrightarrow{e}_{uv}$ and $\overrightarrow{e}_{vu}$, respectively. The sequence numbers of all edges after edge e_{uv} in E_{exgs} increase by one. When only $\overrightarrow{e}_{uv}$ exists in $\overrightarrow{G}$, e_{uv} can be replaced by $\overrightarrow{e}_{uv}$. The sequence number k is assigned to $\overrightarrow{e}_{uv}$. The sequence number of all other edges keep the same. When neither $\overrightarrow{e}_{uv}$ nor $\overrightarrow{e}_{vu}$ exist in $\overrightarrow{G}$, it means that the edge contains at least one pseudo-vertex shrunk from a grid structure. In this case, the edge sets for storing pseudo-vertex mapping relations, E^i_{v2n} and E^{ij}_{v2v}, generated during grid shrinking is required.

When only u is a pseudo-vertex shrunk from grid G^i_{grid} and v is a natural node in G, the edge set E^i_{v2n} is traversed to search the edges connecting grid G^i_{grid} and node v. All the edges linking grid G^i_{grid} and node v will be added to solution set. If there were i edges in edge set E^i_{v2n} containing node v, the edge $\overrightarrow{e}_{uv}$ is replaced by edge $\overrightarrow{e}_{u'_1v}$, $\overrightarrow{e}_{u'_2v}$, $\overrightarrow{e}_{u'_3v}$, ..., $\overrightarrow{e}_{u'_kv}$ and the sequence numbers, k to $k+i-1$ is assigned to them. The sequence numbers of all edges after edge $\overrightarrow{e}_{uv}$ also increase by $i-1$.

When u and v are pseudo-vertices shrunk from grids G^i_{grid} and G^j_{grid} respectively, the edge set E^{ij}_{v2v} is used to find the original edges in graph G. If there were i edges in edge set E^{ij}_{v2v}, the edge $\overrightarrow{e}_{uv}$ is replaced by edge $\overrightarrow{e}_{u'_1v'_1}$, $\overrightarrow{e}_{u'_2v'_2}$, $\overrightarrow{e}_{u'_3v'_3}$, ..., $\overrightarrow{e}_{u'_kv'_k}$ and the sequence numbers, k to $k+i-1$ is assigned to them accordingly. The sequence numbers of all edges after edge $\overrightarrow{e}_{uv}$ increase by $i-1$.

After all the edges in edge set E_{exgs} (solution edge set on shrunk graph) are traversed by the edge mapping process, the E_{exgs} will be updated as $\overrightarrow{E}_{ex}$ (solution edge set on original directed graph). Due to one edge may be replaced by more than one edges in edge mapping produce, $|\overrightarrow{E}_{ex}| \geqslant |E_{exgs}|$. Similarly, the edge set $\overrightarrow{E}_{wst}$ can be fond from the the edge set E_{wstgs}. The procedure to find $\overrightarrow{E}_{wst}$ is identical as Procedure 5, except replacing E_{exgs} by E_{wstgs}.

5.2.3 Algorithm Description of GGM

The algorithm of the Grid-shrinking method is detailed in Algorithm 2. The algorithm starts by initialization. $EC_{\overrightarrow{e}}$ and $WC_{\overrightarrow{e}}$ are initialized as 0 and $\overrightarrow{E}_{ex}$ and $\overrightarrow{E}_{wst}$ are initialized as empty sets. Before the expected contamination degree estimation for each edge, the grid structures on road topology are detected by the Procedure 3. The input graph for grid detection procedure is the same as the input of the Original Greedy Method which is a directed graph representing the road network by the prime approach. The output of grid detection procedure is G_{grid} which is an undirected graph composed of grid structures.

Procedure 5 Edge Mapping

procedure EDGE MAPPING($\overrightarrow{G} = (V, \overrightarrow{E}), G_{grid}, E_{exgs}, \{E^{ij}_{v2v}, E^{i}_{v2n}, i, j \in [1, n]\})$)
  Initialize: $\overrightarrow{E}_{ex} \leftarrow \varnothing, index_{exgs} \leftarrow 0$
  while $index_{exgs} < |E_{exgs}|$ **do**
    edge $e_{uv} \leftarrow E_{exgs}(index)$ ▷ find e_{uv} by its index number
    if $(u \in \overrightarrow{G}) \wedge (v \in \overrightarrow{G})$ **then**
      add e_{uv} to $\overrightarrow{E}_{ex}$ as the last element
    else if $(u \in \overrightarrow{G}) \wedge (v \notin \overrightarrow{G})$ **then**
      $i \leftarrow v$
      for edge $e_{n_1n_2} \in E^{i}_{n2v}$ **do**
        if $(n_1 = u) \vee (n_2 = u)$ **then**
          add $e_{n_1n_2}$ to $\overrightarrow{E}_{ex}$ as the last element ▷ The index number of each newly added element in $\overrightarrow{E}_{ex}$ is increased by one
        end if
      end for
    else if $(v \in \overrightarrow{G}) \wedge (u \notin \overrightarrow{G})$ **then**
      $i \leftarrow u$
      for edge $e_{n_1n_2} \in E^{i}_{n2v}$ **do**
        if $(n_1 = u) \vee (n_2 = u)$ **then**
          add $e_{n_1n_2}$ to $\overrightarrow{E}_{ex}$ as the last element
        end if
      end for
    else
      for $E^{v_1v_2}_{v2v} \in E^{ij}_{v2v}$ **do**
        if $\{(v_1 = u) \wedge (v_2 = v)\} \vee \{(v_1 = v) \wedge (v_2 = u)\}$ **then**
          add all elements in $E^{v_1v_2}_{v2v}$ to $\overrightarrow{E}_{ex}$ as the last $|E^{v_1v_2}_{v2v}|$ element
        end if
      end for
    end if
    $index_{exgs} = index_{exgs} + 1$
  end while
  return E_{ex}
end procedure

Then, the grid shrinking process shown in Procedure 4 is applied. The directed graph $\overrightarrow{G}$ built from road topology is converted into a undirected graph G. G and G_{grid} are the inputs of grid shrinking module. After that, the graph after shrinking is converted into a directed graph and used for the bond percolation threshold estimation and greedy edge ranking. The bond percolation threshold estimation and greedy edge ranking procedures are the same as the procedures shown in Algorithm 1. In the end, the edge mapping module shown in Procedure 5 is applied, which maps the edge set of G_{gs} to the edge set of $\overrightarrow{G}$. For a given input edge budget K_{eb}, the solution sets to minimum expected contamination degree and minimum worst contamination degree can be found by the top K_{eb} edges in $\overrightarrow{E}_{ex}$ and $\overrightarrow{E}_{wst}$, respectively. The edges in $\overrightarrow{E}_{ex}$ are the candidate edges to be the solution to the proposed MVSU deployment problem.

Algorithm 2 Grid-shrinking Greedy Method

Input: $\overrightarrow{G} = (V, \overrightarrow{E}), M, K_p$

Output: $\overrightarrow{E}_{ex}, \overrightarrow{E}_{wst}$

1: **Initialize:** $C_{ex} \leftarrow 0, C_{wst} \leftarrow 0, E_{ex} \leftarrow \varnothing, E_{wst} \leftarrow \varnothing$
2: convert $\overrightarrow{G}$ to G
3: $G_{grid} \leftarrow$ GRID DETECTION$(G = (V, E))$ ▷ G is a undirected graph
4: $G_{gs} \leftarrow$ GRID SHRINKING$(G = (V, E), G_{grid})$ ▷ G and G_{grid} are undirected graphs
5: convert G_{gs} to $\overrightarrow{G}_{gs}$
6: $\overrightarrow{E}_{exgs}, \overrightarrow{E}_{wstgs} \leftarrow$ORIGINAL GREEDY METHOD$(\overrightarrow{G}_{gs}, M, K_p)$
7: $\overrightarrow{E}_{ex} \leftarrow$ EDGE MAPPING$(E_{exgs}, E^{ij}_{v2v}, E^{i}_{v2n})$
8: $\overrightarrow{E}_{wst} \leftarrow$ EDGE MAPPING$(E_{wstgs}, E^{ij}_{v2v}, E^{i}_{v2n})$

5.3 Simplified Greedy Method (SGM)

In the Grid-shrinking Greedy Method, the amount of calculation could be reduced by considering the grid structure as whole. However, the accuracy of grid approximation and the complexity reduction of the grid-shrinking heavily depends on the road topologies. So the road topologies to which the Greedy-shrinking Greedy method could be applied are limited. To overcome these limitations, another road topology independent method is developed. In the new method, instead of running percolation process on a directed graph, the edge direction of the original directed road graph is ignored. As a result, the number of edges used for bond percolation is about half of the original directed graph and the complexity is reduced dramatically. We name this method as Simplified Greedy Method

(SGM) in the following discussion.

5.3.1 Equivalence of Directed and Undirected Graphs

The input graph for Original Greedy Method is a directed ones such that, for any edge $\overrightarrow{e}_{uv} \in \overrightarrow{G}$, there is an edge $\overrightarrow{e}_{vu} \in \overrightarrow{G}$. From a reachability perspective, the directed graph $\overrightarrow{G}$ is equivalent with a undirected graph G', such that an directed edge pair $\{\overrightarrow{e}_{uv}, \overrightarrow{e}_{vu}\}$ is replaced by a undirected edge e_{uv}. The equivalence from a bond percolation aspect is shown by the bond percolation curves on grids in Figure 5.1. For the directed and undirected grid graphs, the bond percolation thresholds are the same. This equivalence inspires us to develop a method employing a corresponding undirected graph to solve the problem on the directed graph.

However, the approach of finding percolation threshold by SCC is not applicable on a undirected graph, so an equivalent estimation method is desired. Based on the definition, the contamination degree of node v_i is the number of reachable nodes from node v_i. For a node v_i in a undirected graph, the number of reachable nodes from node v_i is equal to the node number of the connected component containing node v_i. Instead of finding the reachable nodes of each node, the contamination degree can be fast calculated by the node number of each connected component. All the nodes within the same connected component possess the same number of reachable nodes, e.g. the same contamination degree. The expected contamination degree of a undirected graph can be calculated as Equation 5.7,

$$C_{ex}(G_p^m) = \sum_{j=1}^{J} \sum_{i=1}^{|CC_j|} p_i |CC_j| \tag{5.7}$$

where $CC_j, \{1 \leqslant j \leqslant J\}$ are the connected components in graph G_p^m, and $|CC_j|$ is the node number of CC_j.

The procedure of bond percolation on a undirected graph is almost the same as the one for directed graph. For each edge $e_i \in E$, its appearance is determined by the probability of edge p_{e_i}. The details of bond percolation process for undirected graphs is shown as the Procedure 6.

We are interested in the critical status such that the contamination cannot widely spread on the network. In other words, the giant connected cluster does not exist in G_p. So the connected component is used to estimate the bond percolation threshold of a undirected graph. Similar to the procedure of finding the threshold in a directed graph, the bond percolation threshold of a undirected graph is found by the probability leading to the

Procedure 6 Bound Percolation for Undirected Graphs)

```
1: procedure BOUND PERCOLATION FOR UNDIRECTED GRAPHS(G = (V, E), p_p)
2:     Initialize: G_p = (V, E_p), E_p ← ∅
3:     for all e_i ∈ E do
4:         if p ⩽ p_p then                    ▷ p is a random number, 0 ⩽ p ⩽ 1
5:             E_p ← E_p ∪ {e_i}
6:         end if
7:     end for
8:     return G_p
9: end procedure
```

maximum second largest connected component. The process of finding bond percolation threshold is shown in Procedure 7. In Procedure 7, the bond percolation process is carried out under probability from 0.01 to 0.99 with the step of 0.01. For each probability value p_p, the graph G_p is generated by bound percolation process under percolation probability p_p. The node number of Second Largest Connected Component ($SLCC$) of each G_p is calculated. The bond percolation process at each p_p is repeated K_p times to find the average value of $SLCC$ of each p_p, so the curve of $SLCC$ at each p_p can be drawn. Then the bond percolation threshold p_{th} can be found when $SLCC$ reaches the maximum. Similarly, the Ratio of Cluster Size which is the ratio of node number of the connected component to the total node number of the graph is used to measure the size of largest connected component and $SLCC$. The percolation thresholds curves of corresponding undirected graphs of Map 1 to 6 are shown in Figure 6.4.

5.3.2 Algorithm Description of SGM

When the percolation threshold of the undirected graph is found, the bond percolation procedure of undirected graphs can be used to estimate the expected contamination degree. The details of method three can be found in Algorithm 3. The general steps of Algorithm 3 is similar with Algorithm 1, here we highlight the difference. In the percolation threshold estimation part (shown as Procedure 7), the cluster size is calculated based on the node number of connected component. In line 11, the expected contamination degree, C_{ex}, is calculated based on the number of connected nodes. In the line 22 to line 25, the expected contamination degree and worst contamination degree of each edge e_{uv} are mapped back to the original directed graph as follows. For each directed edge $\overrightarrow{e}_{uv}$ in $\overrightarrow{G}$, the expected contamination degree and worst contamination degree of e_{uv} in G are used for edge ranking. So the expected contamination degree and the worst contamination degree of edge $\overrightarrow{e}_{uv}$

Procedure 7 Bond Percolation Threshold Estimation for Undirected Graphs

1: **procedure** Bond Percolation Threshold Estimation for Undirected Graphs($G = (V, E), K_p$)
2: **Initialize:** $G_p \leftarrow \varnothing, SLCC_{max} \leftarrow 0, p_{th} \leftarrow 0, p_p \leftarrow 0, sl_p \leftarrow 0.01, K \leftarrow 0$
3: **while** $p_p \leqslant 1$ **do**
4: $SLCC_{total} \leftarrow 0$
5: **while** $K < K_P$ **do**
6: $G_p \leftarrow$ Bound Percolation for Undirected Graphs($G = (V, E), p_p$)
7: $SLCC_{total} \leftarrow SLCC_{total} + SLCC$
8: $K \leftarrow K + 1$
9: **end while**
10: $\overline{SLCC} = SLCC_{total}/K_p$
11: **if** $SLCC_{max} < SLCC$ **then**
12: $SLCC_{max} \leftarrow SLCC$
13: $p_{th} \leftarrow p_p$
14: **end if**
15: $p_p \leftarrow p_p + sl_p$
16: **end while**
17: **return** p_{th}
18: **end procedure**

will be the same as edge $\overrightarrow{e}_{vu}$. The top K_{eb} edges in edge set $\overrightarrow{E}_{ex}$ and $\overrightarrow{E}_{wst}$ are used as the final solution set to the minimum expected contamination degree and the minimum worst contamination degree problems. The edge list $\overrightarrow{E}_{ex}$ is used as the solution to MVSU deployment problem on road network for the given K_{eb}.

Algorithm 3 Simplified Greedy Method

Input: $G = (V, E), M, K_p$

Output: $\overrightarrow{E}_{ex}, \overrightarrow{E}_{wst}$

Initialize: $C_{ex} \leftarrow 0, C_{wst} \leftarrow 0, E_{ex} \leftarrow \varnothing, E_{wst} \leftarrow \varnothing$
for all $e_i \in G$ **do**
    $EC_{e_i} \leftarrow 0$
    $WC_{e_i} \leftarrow 0$
    $counter_{e_i} \leftarrow 0$
end for
$p_{th} \leftarrow$ Percolation Threshold Estimation for Undirected Graphs($G = (V, E), K_p$)
while $m < M$ **do**
    $G_p^m \leftarrow$ Bound Percolation for Undirected Graphs($G = (V, E), p_{th}$)
    for all $v_i \in G_p^m$ **do**
        $C_{ex} \leftarrow C_{ex} + P(v_i) * C(v_i)$ ▷ C_{ex} is calculated based on connected node number
        if $C_{wst} < C(v_i)$ **then**
            $C_{wst} \leftarrow C(v_i)$
        end if
    end for
    for all $e_i \in G$ and $e_i \notin G_p^m$ **do**
        $EC_{e_i} \leftarrow (EC_{e_i} + C_{ex})$
        $WC_{e_i} \leftarrow (WC_{e_i} + C_{wst})$
        $counter_{e_i} \leftarrow (counter_{e_i} + 1)$
    end for
end while
for all $\overrightarrow{e}_i \in \overrightarrow{G}$ **do**
    $\overline{EC}_{\overrightarrow{e}_i} \leftarrow (EC_{e_i}/counter_{e_i})$
    $\overline{WC}_{\overrightarrow{e}_i} \leftarrow (WC_{e_i}/counter_{e_i})$
end for
$\overrightarrow{E}_{ex} \leftarrow sorting : \overrightarrow{E}$ *by* $\overline{EC}_{\overrightarrow{e}_i}$
$\overrightarrow{E}_{wst} \leftarrow sorting : \overrightarrow{E}$ *by* $\overline{WC}_{\overrightarrow{e}_i}$

Chapter 6

Experimental Results

The performance of three proposed methods is evaluated by applying them to real road networks. The max-flow based method proposed in [7] is implemented as the comparison method. The three proposed greedy methods and the comparison method are discussed regarding the scalability and complexity, contamination degree, and blocking probability.

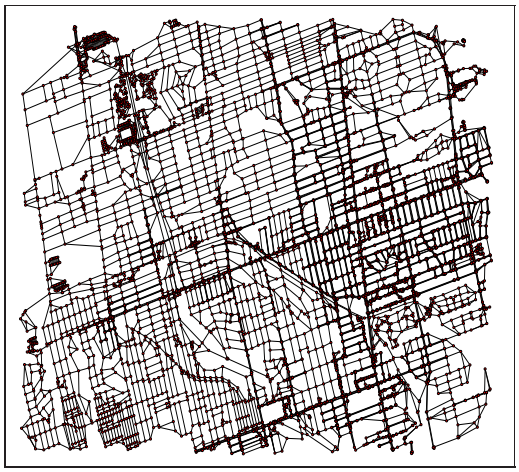
The robustness of the maximum flow based method in [7] has been shown in [93] by applying it to 50 cities in the United States. In this approach, a directed graph is built based on a selected annulus area which is centered at the central area of the city. The inner and outer radiuses of the annulus area are predefined. The annulus area is named as the "buffer area". The max-flow of the buffer area is found by introducing two auxiliary nodes, a virtual source node, and a virtual sink node. All the edges cut by the inner radius are connected to a virtual source and all the edges cut by the outer radius are connected to a virtual sink. The minimum cut between the virtual source and sink are found by solving the maximum flow problem on road network in the buffer area between the nodes of source and sink.

6.1 Simulation Environment

Our simulation is implemented on real road networks acquired from Open Street Map [47]. Six areas are selected from the map data of Ottawa, Toronto and New York City. The corresponding graphs are built based on the approach introduced in Section 4.2.1. The graphs of map 1 to 6 built from map data are shown in Figure 6.1. The proposed methods and the comparison method are implemented in Python and the graphs are built by the NetworkX [55].

(a) Map 1 [53]

(b) Map 2 [50]



(c) Map 3 [52]

(d) Map 4 [51]

(e) Map 5 [48]

(f) Map 6 [49]

Figure 6.1: Graphs Built from Map Data

Table 6.1: Summary of Road Graphs Used for Simulation

Map Name	City	Road Graph (Directed)		Road Graph After Grid Detection		Road Graph (Undirected)	
		Node Number	*Edge Number*	*Node Number*	*Edge Number*	*Node Number*	*Edge Number*
Map 1	Ottawa	2460	8186	1411	4742	2460	4093
Map 2	Ottawa	2610	8188	2166	6824	2610	4094
Map 3	Toronto	8315	28144	3369	11660	8315	14072
Map 4	Toronto	7588	25414	3698	12596	7588	12707
Map 5	New York	8246	29312	2468	8492	8246	14656
Map 6	New York	6101	21416	2051	7198	6101	10708

6.2 Experimental Settings

We assume the potential location of the initial infectious vehicle follows the uniform distribution. The nearest intersection is used to approximate the location of the initial infectious vehicle, so each intersection has an equal probability to be the contamination source node. The advantage of employing the uniform distribution is that the values of expected and average contamination degree would be the same so that the Original Greedy method would be exactly the same as the method proposed in [70]. As a result, the Original Greedy method could be used as the baseline method to access the performance of the Grid-shrinking Greedy Method and Simplified Greedy Method on road networks.

According to the principle of bond percolation approximation, a sufficient large iteration number M is a required to estimate the contamination degrees. However, the amount of calculation increases along with the increase of M. So we are looking for a proper value of M such that M is large enough to make the solution set converging, and M should be as small as possible to avoid unnecessary calculation complexity. Kimura et al. [70] argued that several thousand iterations would be sufficiently large for contamination degree estimation on a scale-free network. The total iteration number, M, was estimated by Equation 6.1 [70],

$$M = \frac{M'}{1-p} \tag{6.1}$$

where M' is the expected valid iterations number for contamination degree estimation. So we first apply the bond percolation threshold estimation module to find the uniform percolation probability. The graphs shown in Figure 6.1 are used as input graphs for the Original Greedy method and the bond percolation threshold estimation. We set K_p as 100 for all graphs. The bond percolation threshold estimation curves of the graphs in Figure 6.1 are shown in Figure 6.4.

Although the graphs in Figure 6.1 are also used as input for the Grid-shrinking Greedy Method and Simplified Greedy Method, the graphs used for bond percolation threshold

(a) Map 1

(b) Map 2

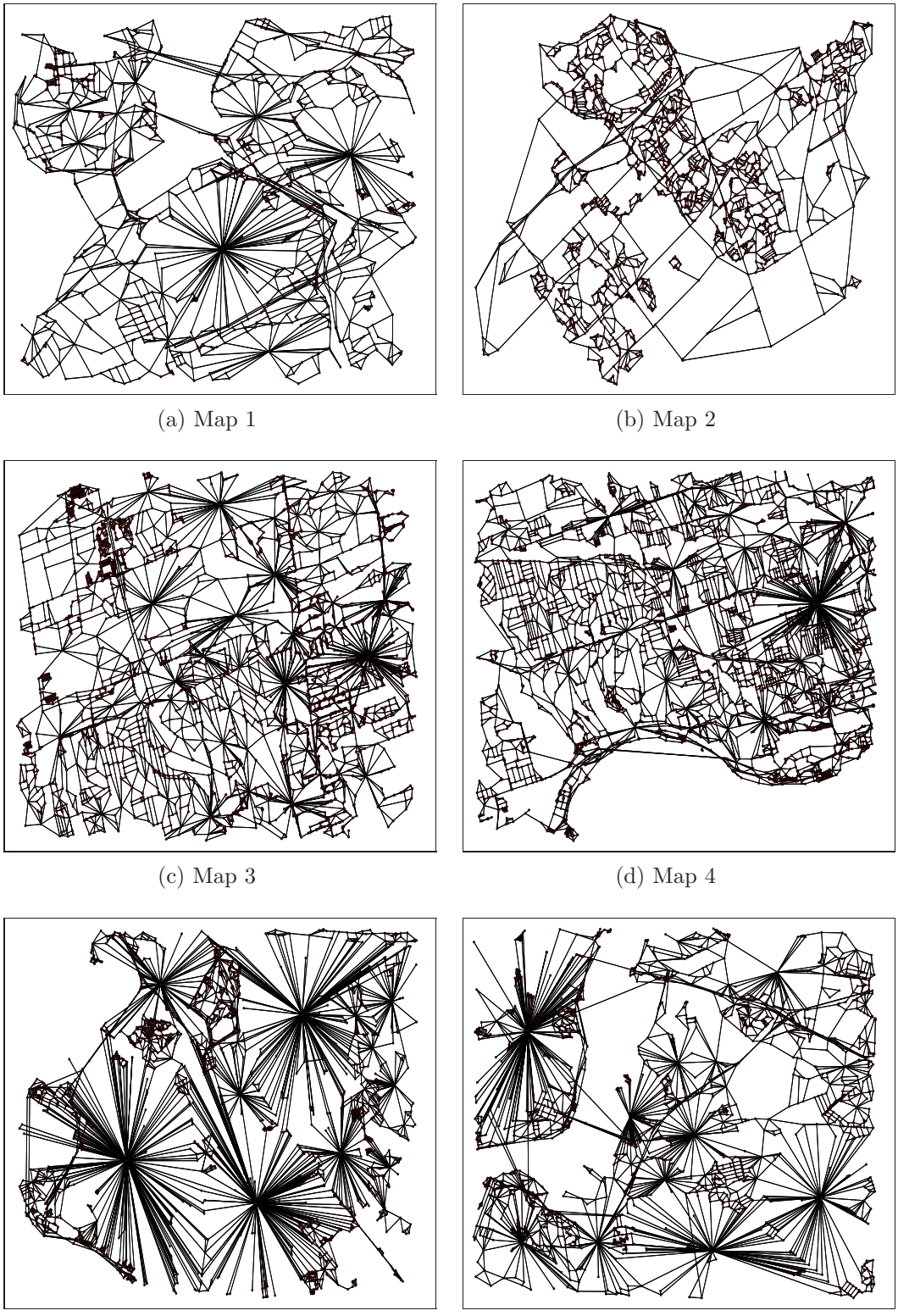
(c) Map 3

(d) Map 4

(e) Map 5

(f) Map 6

Figure 6.2: Grid Detection Results of Map 1 to 6



(a) Map 1

(b) Map 2

(c) Map 3

(d) Map 4

(e) Map 5

(f) Map 6

Figure 6.3: Grid Shrinking Results of Map 1 to 6

Table 6.2: Percolation Threshold and M for Each Method

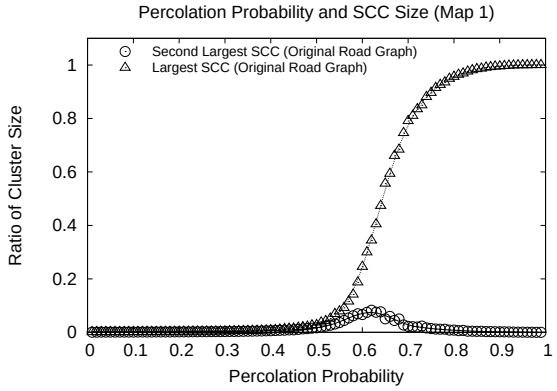
Map	**Expected M'**	Pt_{OGM}	M_{OGM}	Pt_{GGM}	M_{GGM}	Pt_{SGM}	M_{SGM}
1	10000	0.62	26315	0.55	22222	0.59	24390
2	10000	0.71	34482	0.69	32258	0.66	29411
3	10000	0.59	24390	0.45	18181	0.59	24390
4	10000	0.6	25000	0.58	23809	0.61	25641
5	10000	0.57	23255	0.66	29411	0.55	22222
6	10000	0.58	23809	0.49	19607	0.57	23255

estimation are different from the Original Greedy method. According to the Grid-shrinking Greedy Method shown in Algorithm 2, the grid detection module and grid shrinking module are applied to the input graph before the bond percolation process. The grid detection and grid shrinking results of map 1 to 6 are shown in Figure 6.2 and 6.3, respectively.

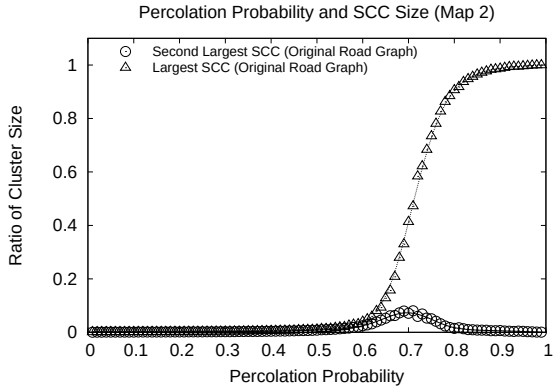
The graphs after grid-shrinking shown in Figure 6.3 are used for the bond percolation threshold estimation in Grid-shrinking Greedy Method. For the Simplified Greedy Method, the corresponding undirected graphs of the graphs in Figure 6.1 are used to find their percolation threshold. The same K_p ($K_p = 100$) is used to estimate the bond percolation threshold of each graph. The curves of bond percolation threshold estimation from the Original Greedy method, Grid-shrinking Greedy Method, and Simplified Greedy Method are compared in Figure 6.7 which will be further discussed in Section 6.4.

The approximate bond percolation thresholds and corresponding iteration numbers are listed in Table 6.2, where the iteration numbers are calculated by Equation 6.1. Column Pt_{OR} and M_{OGM} are the bond percolation threshold and bond percolation iteration number used for the Original Greedy method. Similarly, Pt_{GGM}, M_{GGM}, Pt_{SGM}, and M_{SGM} are the columns of percolation thresholds and iteration numbers for Grid-shrinking Greedy Method (GGM) and Simplified Greedy Method (SGM), respectively.

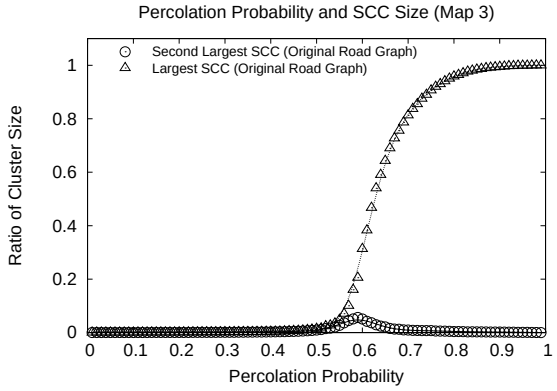
To implement the comparison method (the max-flow based method [7]), we use the center of the maximum inscribed circle of each map to be the center of buffer area. The inner and outer radius of the buffer area are governed by the ratio to the radius of maximum inscribed circle. For example, $inner\ radius = 0.3$ and $outer\ radius = 0.6$ mean that the length of inner radius is $0.3 * R_{max}$ and length of outer radius is $0.6 * R_{max}$, where R_{max} is the radius of maximum inscribed circle. It is found that the edge cut numbers are dependent of inner radius. The edge cuts found by the max-flow based method with different radius values are shown in Table 6.3. The edge cuts are used as K_{eb} for the proposed methods. Note that the three proposed methods could find the solution edge set for any given positive integer K_{eb}. However, the size of a solution set from the max-flow based method cannot be predefined. So we first find the edge cuts with different inner and outer radius values



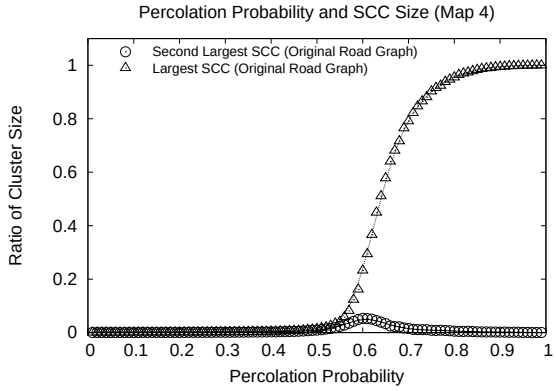
(a) Map 1



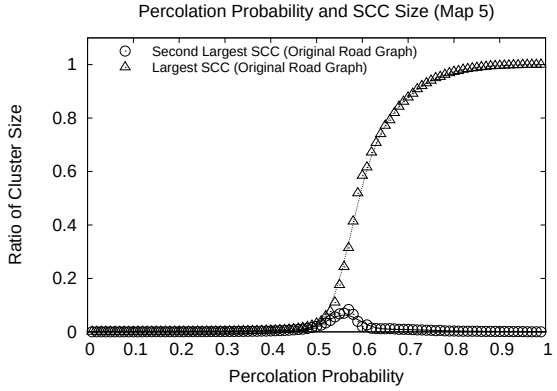
(b) Map 2



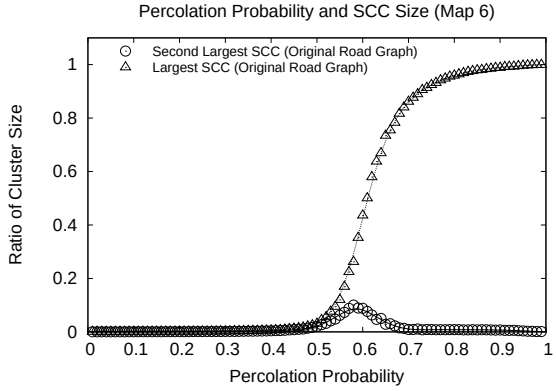
(c) Map 3



(d) Map 4



(e) Map 5



(f) Map 6

Figure 6.4: Bond Percolation Threshold (Directed Graphs)

Table 6.3: Edge Cut Found by Max-flow Method

Inner Radius	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Edge Cut Numbers (K) of Map 1	32	52	68	84	102	116	128	130
Edge Cut Numbers (K) of Map 2	34	40	46	50	62	66	76	84
Edge Cut Numbers (K) of Map 3	34	74	106	126	146	180	204	246
Edge Cut Numbers (K) of Map 4	38	52	72	110	120	128	158	192
Edge Cut Numbers (K) of Map 5	52	118	160	198	254	306	370	452
Edge Cut Numbers (K) of Map 6	66	66	68	76	78	96	134	200

by the max-flow based method then use the edge cuts as the input K_{eb} for the Original Greedy Method, Grid-shrinking Greedy Method, and Simplified Greedy Method.

6.3 Scalability and adaptability

In the three proposed greedy based methods, bond percolation significantly reduces the amount of calculation. It is notable that in Table 6.2, the total iterations required for percolation M is independent of the size of the graphs. According to Equation 6.1, the necessary iteration number only relies on the percolation threshold of the graph. For each percolation output graph G_p, the edge number of G_p is approximate $p_t * |E|$, where p_t is the percolation threshold used as the percolation probability and $|E|$ is the edge number of original input graph. The complexity of finding strongly connected component in a digraph is $O(|E| * |V|)$ which is the same as the complexity of finding connected component in a undirected graph. Therefore, the complexity of the three greedy based methods can be expressed by Equation 6.2.

$$O(M * |E|^2 * |V| * p_t) \tag{6.2}$$

In the comparison method, the Ford−Fulkerson method is used in [5] to find the minimum edge cut in the buffer area. The complexity of standard Ford−Fulkerson method is $O(C * E')$, where C is the capacity and E' is the total edge number. Note there the E' are the edges in the auxiliary graph for solving the max-flow problem. E' depends on the size of buffer area and the edges being cut by the inner and outer radius. The capacity used for modeling the buffer area as a max-flow problem is usually a small number, so the influence of C on the complexity is negligible.

The max-flow method finds the minimum edge cut, but the edge number of the edge cut is unpredictable. As a result, the max-flow method cannot work for an arbitrary input

edge budget K_{eb}. Both the Original Greedy method and the Simplified Greedy Method can find a solution set for an arbitrary input edge budget K_{eb}. Moreover, the these two methods can get a solution for any $1 \leqslant K_{eb} \leqslant |E|$ after one-time execution. Therefore, the adaptability of these two proposed methods is the better than the comparison method. The applicable edge budget range of the Grid-shrinking Greedy Method is dependent of the input graph because the edges within the grid structures cannot be selected into solution set. The applicable edge budget range can be expressed by $1 \leqslant K_{eb} \leqslant 2|E_{gs}|$, where $|E_{gs}|$ is the edge number of G_{gs} (the graph after grid shrinking). Because G_{gs} is an undirected graph, the edge number will be doubled when it is converted into a directed graph.

The Grid-shrinking Greedy Method(GGM) and the Simplified Greedy Method(SGM) are developed based on the Original Greedy Method(OGM). The graph size for percolation is reduced by grid shrinking and graph converting, so the scalability and complexity of proposed Grid-shrinking Greedy Method and the Simplified Greedy are less than the Original Greedy method. The grid shrinking process creates a number of pseudo-vertices. We define graph G_{gs} is a graph G after grid shrinking, v_i^{pv} is a pseudo-vertex shrunk from grid G_{grid}^i. According to the definition of expected contamination degree (Equation 4.4), the probability that v_i^{pv} is selected as contamination source node in G_S is the sum of $p(v_i)$ of node v_i, where v_i is a node in grid G_{grid}^i and $p(v_i)$ is the probability of v_i being selected as source node. By this shrinking process, the graph size could be a greatly reduced, so it significantly reduces the amount of calculation. The graph size reduction of each map can be found in the Table 6.1. The accuracy and the complexity reduction of grid shrinking approximation are dependent of the road graph characteristics (the ratio and size of the grid structures). In general, the Grid-shrinking Greedy Method has a good approximation when the edge budget is much less than the edge number required to fairly disconnect a graph into several disconnected graphs.

The six real networks are used to investigate the complexity reduction by the GGM and SGM. We calculate the complexity by Equation 6.2 and compare the complexity of OGM, GGM and SGM to the complexity of OGM so that the complexity of OGM is always 100%. The result is shown in Figure 6.5. Based on the result, the complexity of SGM is about 20% of the OGM and the complexity of GGM is various from 5% to 50% of the OGM. It can be found that although GGM could make a greater complexity reduction than the SGM, the amount of complexity reduction is independent of the input graphes.

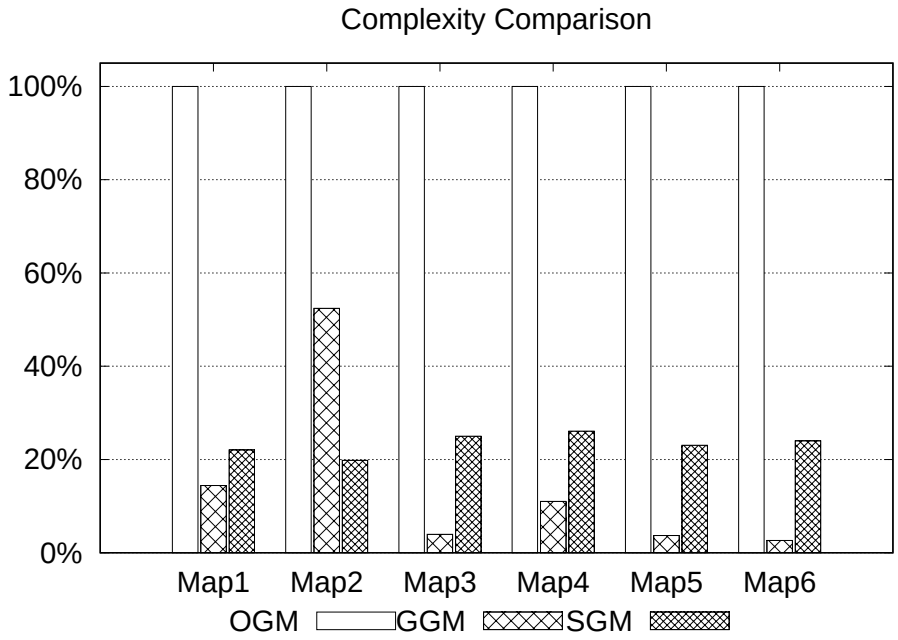


Figure 6.5: Complexity Comparison

6.4 Degree Distribution and Percolation Threshold

The Grid-shrinking Greedy Method utilizes the road topology pattern to reduce the complexity. However, the grid shrinking process could significantly influence on the characteristics of the graph used for bond percolation. In this section, the graph characteristics before and after grid shrinking are examined regarding graph size, degree distribution, and percolation threshold.

The grid shrinking reduces the graph size remarkably. The decrease of graph size can be found in Table 6.1. Because the complexity of bond percolation mainly depends on the edge number, the reduction of graph size is accessed by the edge number. The size of Map 1 is reduced around 42% after grid shrinking. The edge number of Map 2 are reduced around 17%. The reduction of edge number of Map 3, 4, 5 and 6 are more than 50%. The reduction of graph size reduces the complexity of each percolation iteration. When M is a sufficiently large number, the reduce of total complexity is remarkable.

The influence of grid shrinking to the graph topology which can be observed by degree distributions. Road networks are spatial networks that limited by nature and artificial landscape. The degrees of nodes in road networks are constrained by the physical access of road sections. The most common traffic intersections are “cross” and “T” intersections, which reflects in the degree distributions of road networks. The degree distributions of the graphs before and after grid shrinking are shown in Figure 6.6. In the original graphs from

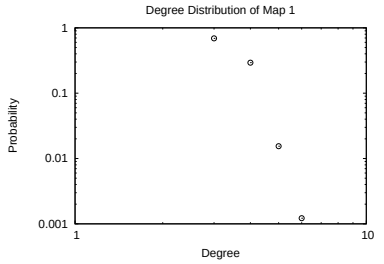
map data (without grid shrinking), it can be found that the numbers of nodes with degree 3 and 4 take a dominate number among all six maps. Grid shrinking process creates pseudo-vertices by shrinking grid structures, so the degree distributions after greedy shrinking are different from the original road graphs. The degree values after grid shrinking are diverse compared with the degree values of original graphs. A number of high degree vertices are created. The degree distributions of graphs after grid shrinking somehow follow the power-law. The Original Greedy method is designed for social networks, and social networks are scale-free networks with power-law degree distribution. Therefore the greedy method still could be applied to the graphs after grid shrinking.

The graph converting is an approximation method to reduce the complexity. The topology features of the original should be kept as much as possible to ensure that the solution set would also be valid in the original graph. The percolation threshold is an eigenvalue of the graph which is determined by the graph characteristic. Therefore, the bond percolation threshold could also be used to access with what extent the graph has been modified from a topology aspect. The three curves of bond percolation thresholds for the original directed graph, the directed graphs after grid shrinking and the undirected graph are drawn and compared. The comparison of the percolation thresholds of map 1 to 6 can be found in Figure 6.7. According to Figure 6.7a and 6.7b, the trends of the three bond percolation threshold curves in map 1 and 2 are similar. It means the graph after grid-shrinking still keep the topology feature of the original graphs. However, in Figure 6.7d, 6.7e, and 6.7f, the curves of graphs after shrinking become more smooth compared with the graphs from original road networks which means the vanishing of the critical state. A less distinct critical points is an important characteristic of the scale-free networks, which also shows a graph after grid shrinking possesses the feature of the scale-free networks.

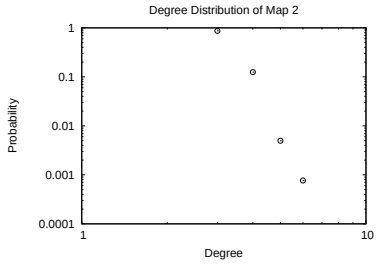
Based on the curves of bond percolation threshold of in Figure 6.7, it can be concluded that the grid-shrinking alters the topology features of the graphs of map 3, 4, 5, and 6. The impact of the grid-shrinking on the topology features may affect the performance of the Grid-shrinking Greedy Method because the graph used to find a solution does not hold the topology features of the original graph.

6.5 Contamination Degree

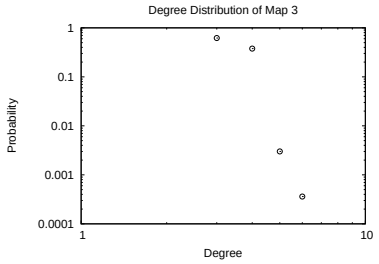
The expected and worst contamination degrees are used to compare the performance of the three proposed methods. A series of increasing edge budgets are used to find the solution sets by applying the three proposed methods. The contamination degree is calculated



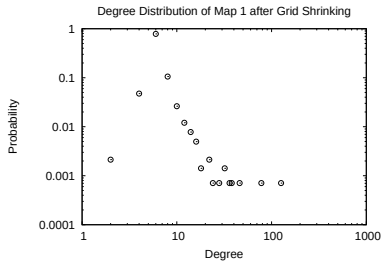
(a) Map 1



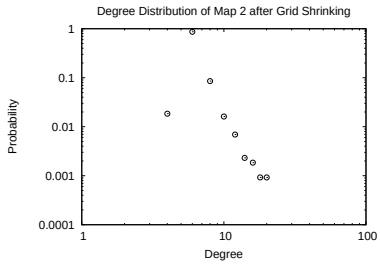
(b) Map 2



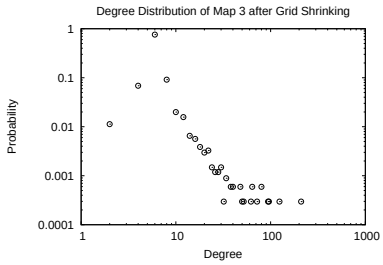
(c) Map 3



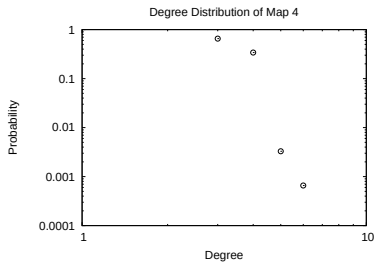
(d) Map 1 After Grid Shrinking



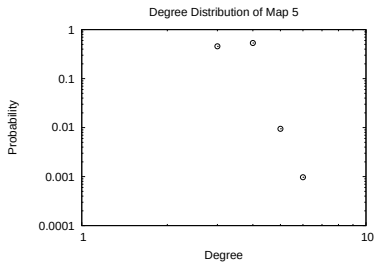
(e) Map 2 After Grid Shrinking



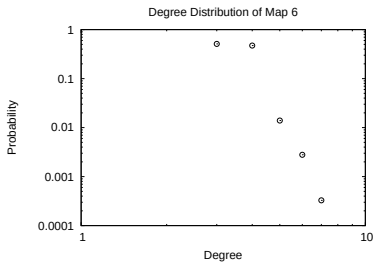
(f) Map 3 After Grid Shrinking



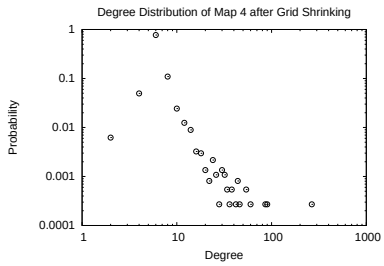
(g) Map 4



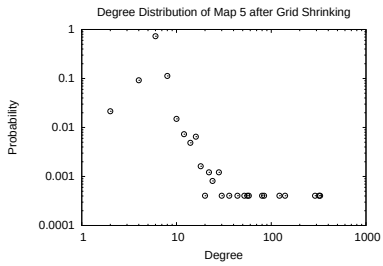
(h) Map 5



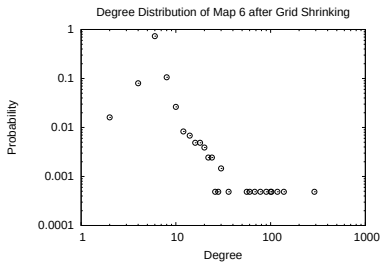
(i) Map 6



(j) Map 4 After Grid Shrinking

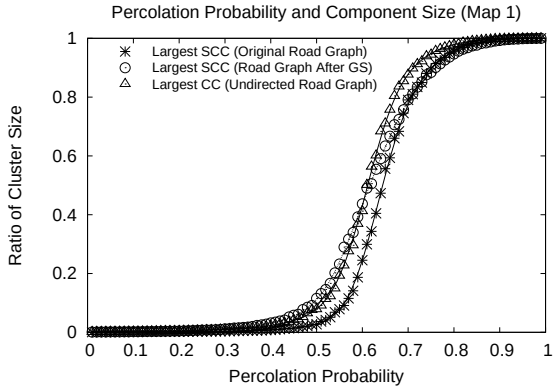


(k) Map 5 After Grid Shrinking

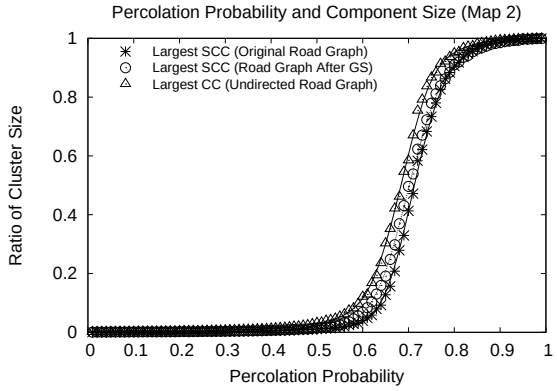


(l) Map 6 After Grid Shrinking

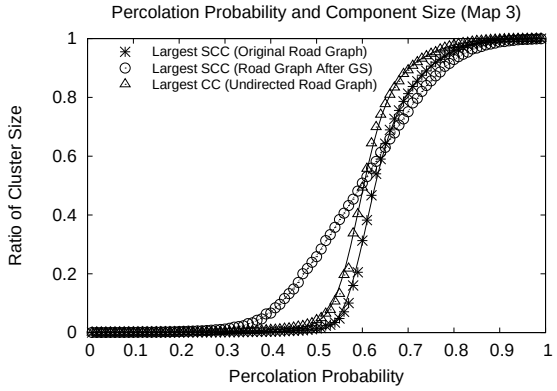
Figure 6.6: Degree Distribution before and after Grid Shrinking



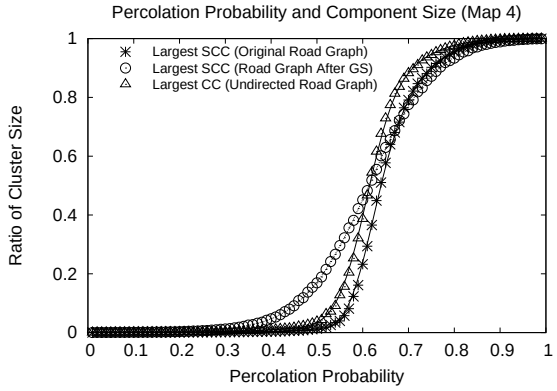
(a) Map 1



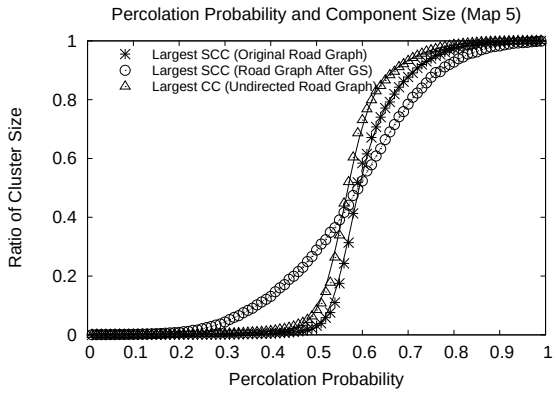
(b) Map 2



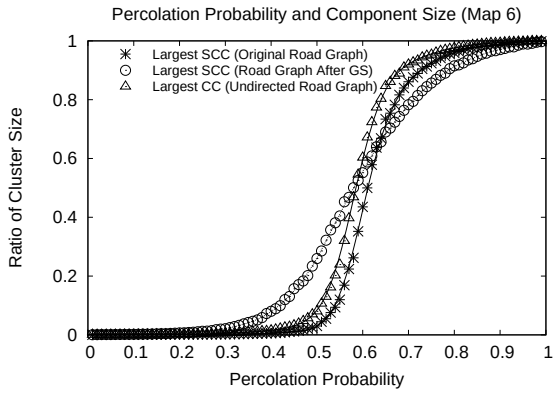
(c) Map 3



(d) Map 4



(e) Map 5



(f) Map 6

Figure 6.7: Bond Percolation Threshold Curves Comparison

by the reachable nodes. The expected and worst contamination degrees are found by the Equation 4.4 and 4.3 respectively. For an input graph and a given edge budget, the solution set leading to a smaller value of contamination degree is better.

The diffusion process under IC model can be imitated by bond percolation, so we still use bond percolation to estimate the value of contamination degree. Note that the bond percolation iterations used for result evaluation are separate from the iterations used to find the solution sets. For each directed graph built from map data, the percolation threshold is found by the process shown in Procedure 2. Then, for a given edge budget K_{eb}, the edge sets E_{ex} and E_{wst} from the proposed methods are evaluated individually. The selected edge set E_{ex} is removed from the directed graph before the bond percolation process. After that, bound percolation process are applied to graph $\vec{G}\backslash E_{ex}$ by sufficient iterations to estimate the expected contamination degree of the graph. Similarly, the selected edge set E_{wst} is removed from the directed graph before the bond percolation process. After that, bound percolation process are applied to graph $\vec{G}\backslash E_{wst}$ by sufficient iterations to estimate the worst contamination degree of the graph. In our simulation, the bond percolation iteration number used for contamination degree estimation is 1000. The range of K_{eb} is from 100 to 1000 with step length 100. As a result, for every input map, there would be 10 solution sets to Minimum Expected Contamination Degree (MECD) problem and 10 solution sets to Minimum Worst Contamination Degree (MWCD) problem.

For each input graph, the contamination degree curves are employed to illustrate the reduction of contamination degree. The curves are drawn by the different edge budgets and the corresponding contamination degrees. For the same edge budget, the solution sets to MECD problem and MWCD problem found by the proposed Original Greedy Method (OGM), Grid-shrinking Greedy Method (GGM), and Simplified Greedy Method (SGM) are compared. We would like to observe the extent of contamination degree reduction by blocking the particular edge set selected by the method under evaluation. For every edge budget value, the solution set leading to the lower contamination degree means the more effective for contamination degree reduction. In other words, from a worm propagation aspect, the solution set leading to the lower value of contamination degree stands for the greater capability for blocking worm propagation.

The expected contamination degree and worst contamination degree of the solutions from the three methods are drawn in the same figure for comparison. The simulation results of all the six maps can be found in Figure 6.8. According to the results, both the expected and worst contamination degree could be reduced by blocking a number of edges. The values of expected and worst contamination degree decrease with the increasing edge budget. As shown in Figure 6.8a, 6.8b, 6.8c, and 6.8f, the solution set found by Gird-shrinking Greedy method and Simplified Greedy Method leading to a lower contamination

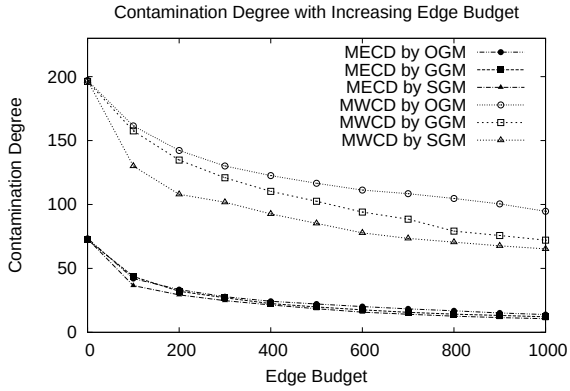
degree compared with the solution set found by the Original Greedy method. Based on Figure 6.8d and 6.8e, the contamination degrees by the solution sets from Grid-shrinking Greedy Method are greater than the Original Greedy method, but the solution sets found by Simplified Greedy Method still lead to the lowest contamination degrees.

To better understanding the solution found by different methods, we use the Original Greedy method as baseline method to evaluate the solution sets found by the Gird-shrinking and Simplified Greedy Method. For the same edge budget, let C_{OGM}, C_{GGM}, and C_{SGM} stand for the contamination degree based on the solution from the Original Greedy method, Gird-shrinking Greedy method, and Simplified Greedy Method, respectively. The equation we used to normalize the expected and the worst contamination degree are shown in Equation 6.3.

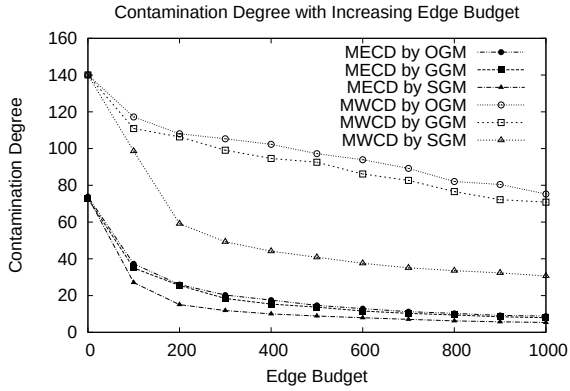
$$R_{SUE} = \frac{C_{SUE} - C_{OGM}}{C_{OGM}} \tag{6.3}$$

where the R_{SUE} is the ratio of the solution under test to the solution found by the baseline method, C_{SUE} is the estimated contamination degree by the solution under evaluation. The C_{SUE} is replaced by C_{OGM}, C_{GGM}, and C_{SGM} correspondingly to normalize the evaluation results of different methods. Both the solutions to MECD problem and to MWCD problem are normalized and compared. The comparisons of the solution to MECD problem are shown in Figure 6.9. Figure 6.10 shows the comparison of the solutions to MWCD problem found by each method.

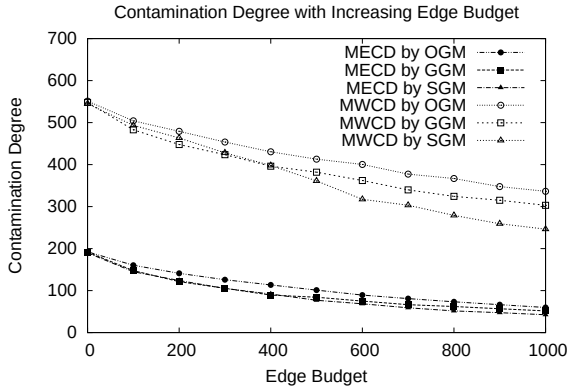
According to Figure 6.9a, 6.9b, and 6.9c, the expected contamination degrees by the solution from Grid-shrinking method is about 10% to 15% less than the baseline solution on average. In Figure 6.9f, the solution from Grid-shrinking Greedy Method is better than the baseline solution when the edge budgets are relatively small. When edge budget increases, the solutions found by the Original Greedy method and Grid-shrinking Greedy Method are comparable. However, on the map 4, and 5, the performance of Grid-shrinking Greedy Method is not as good as other methods, which can be observed in Figure 6.9d and 6.9e. In Figure 6.9d and 6.9e, when edge budget is greater than 400, the expected contamination degree by the solution from the Grid-shrinking Greedy Method is about 20% more than the baseline solution. Compared with the Grid-shrinking Greedy Method, the performance of the Simplified Greedy Method is stable for all maps. The expected contamination degree by the solution from the Simplified Greedy Method is about 20% less than the baseline solution on average for all six maps. The advantage of the Simplified Greedy Method is more evident when the edge budget is increased. For example, the solution set found by the Simplified Greedy Method with edge budget 100 is about 10% better than the baseline method for all maps except map 6. When the edge budgets increase to 600, as shown in



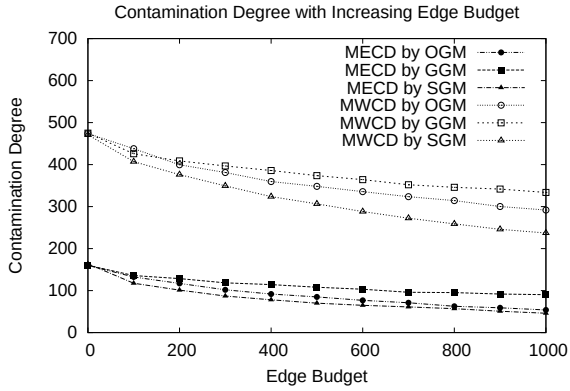
(a) Map 1



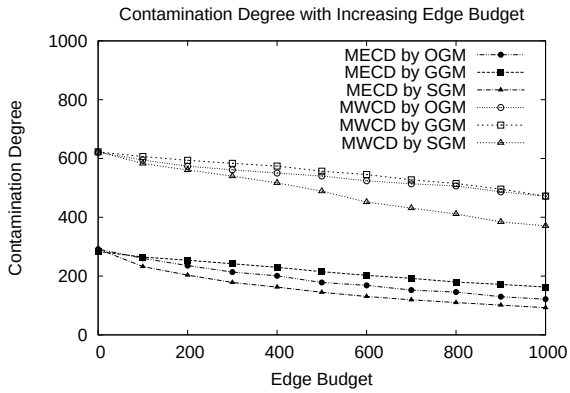
(b) Map 2



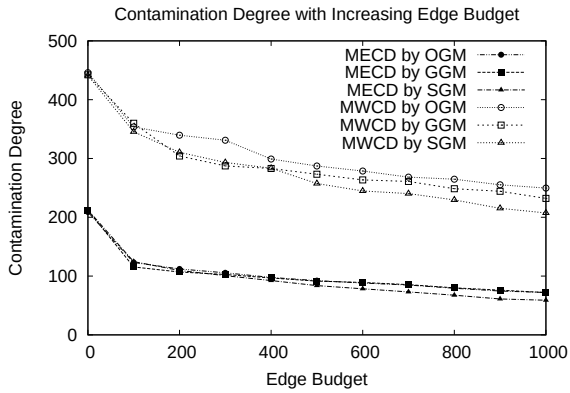
(c) Map 3



(d) Map 4



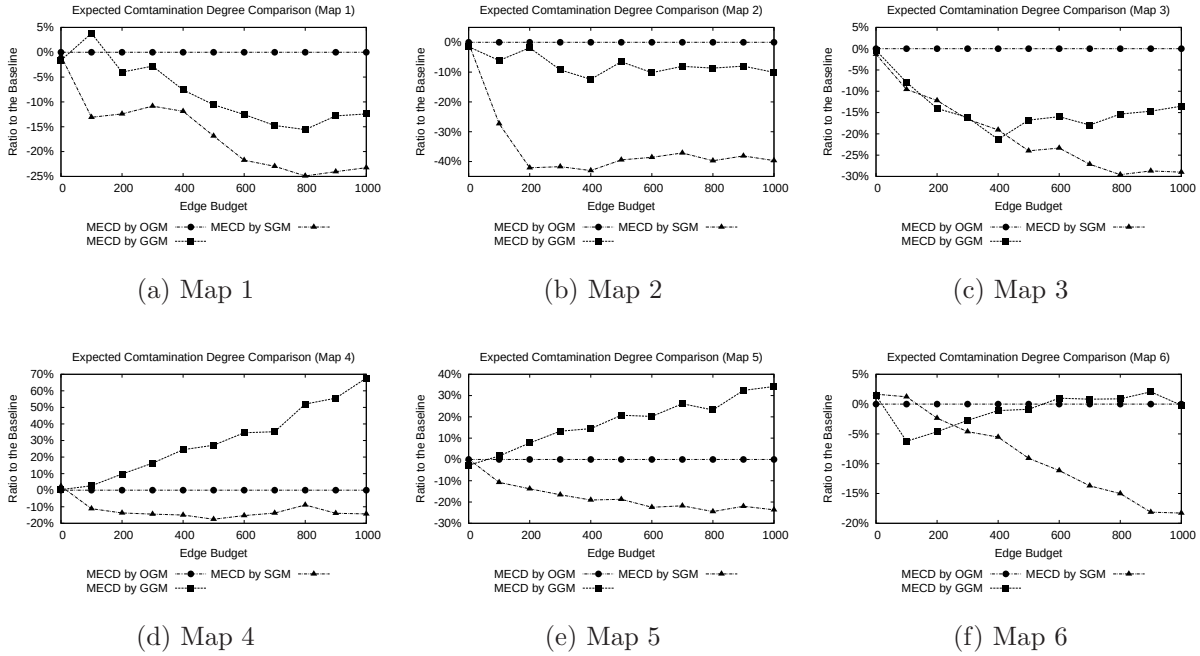
(e) Map 5



(f) Map 6

Figure 6.8: Result Evaluation by Contamination Degree

Figure 6.9a, 6.9b, 6.9c, and 6.9e, the expected contamination degrees by solutions from Simplified Greedy Method are at least 25% less than the baseline method.



(a) Map 1 (b) Map 2 (c) Map 3

(d) Map 4 (e) Map 5 (f) Map 6

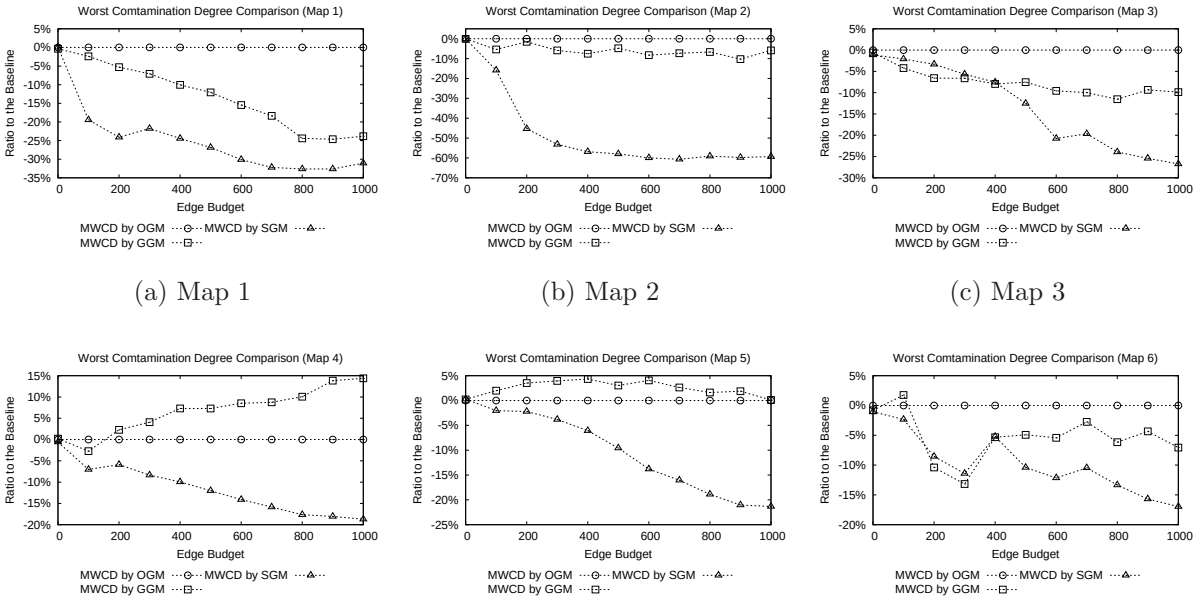
Figure 6.9: Solution Comparison (Solutions to MECD)

Similar conclusions can be found in the solution sets to the minimum worst contamination degree problem. As shown in Figure 6.10, the solution sets from the Greedy-shrinking and Simplified Greedy Method leading to a significant lower worst contamination degree than the baseline method on the map 1, 2, 3, and 6. In map 4 and 5, deterioration can be observed in the solution sets from the Greedy-shrinking method, but the Simplified Greedy Method still outperforms other two methods.

Based on the diagrams in the Figure 6.9 and 6.10, in contrast to the robust performance of the Simplified Greedy Method, the performance of the Grid-shrinking Greedy Method worsens on map 4 and 5. Based our previous analysis in Section 6.3, the grid shrinking alters graph characteristics which may lead to an adaptability issues on some road topologies. The simulation results given by contamination degree agree on the analysis conclusion.

6.6 Average Blocking Probability

We employ the shortest step path (SSP) to represent both the possible V2V communication routing paths and physical routes of vehicles. The average blocking probability p_b of the



(a) Map 1 (b) Map 2 (c) Map 3

(d) Map 4 (e) Map 5 (f) Map 6

Figure 6.10: Solution Comparison (Solutions to MWCD)

solutions from different methods are compared based on the top K shortest step paths between a number of random selected source and destination node pairs. The source node and destination node of each path are selected randomly such that an infectious node s is randomly chosen on the graph, and then a destination t is randomly selected in the rest nodes. After that, the top K shortest step paths between node s and t could be found by Yen's K-shortest path algorithm [97]. This procedure is repeated I iterations, and then a statistic probability can be determined to indicate the probability that a mobile target (could be a vehicle or an infectious message) would be blocked when moving from node s to node t. Let p_b^{ss} denote the average blocking probability of shortest step paths. p_b^{ss} can be calculated by Equation 6.4.

$$p_b^{ss} = \frac{I_b^{ss}}{KI^{s,t}} \tag{6.4}$$

where I_b^{ss} is the total number of shortest step paths that there is at least one MVSU on the path, $I^{s,t}$ is the total number of s,t node pairs and K is number of shortest paths adopted in assessment for each s,t pair. In our simulation, $I^{s,t}$ is set as 5000 and K is set as 10 for all six maps.

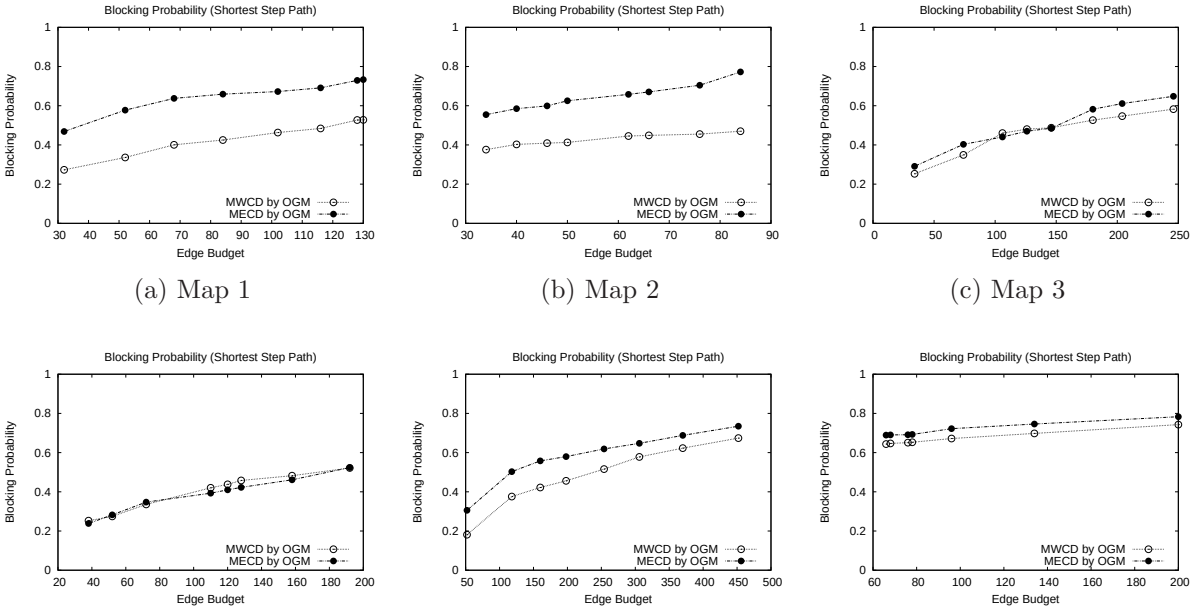
6.6.1 Solutions Found by MECD and MWCD

We modeled the worm spreading problem as minimum contamination problem on road networks. Instead of the worst contamination degree, the expected contamination degree is used to quantify the optimization objective. According to the definition, the expected contamination degree describes the expected number of nodes could be impacted by a single source node. The worst contamination degree refers the maximum number of nodes could be impacted by a single source node. The expected contamination minimization problem and worst contamination minimization problem are two independent problems and their solution sets are also independent. We argue that the optimization objective of expected contamination minimization problem would be better matching the worm containment objective. By simulation, we would like to show that the solution to MECD problem outperforms the solution to MWCD problem from the worm blocking aspect.

Figure 6.11 shows the shortest step path blocking probability of the edges sets to the MECD problem and to the MWCD problem selected by the Original Greedy Method (OGM). It can be found that the blocking probabilities of the solution to MECD problem are insistent greater than the solution to MWCD problem. In map 1 and 2, the blocking probability of solution to MECD problem is about 0.2 higher than the solution to MWCD problem on average. In map 3, 4, 5, and 6, although the advantage of the solution to MECD problem is reduced, the blocking probabilities of the solutions to MECD problem are still greater than the solutions to MWCD in general. Similar results can be found by the solution sets from the proposed Grid-shrinking Greedy Method (GGM) and the Simplified Greedy Method (SGM) shown in Figure 6.12 and Figure 6.13. It is notable that the blocking probabilities of the solution to MECD problem and solution to MWCD problem found by Simplified Greedy Method (SGM) are very close to each other, but the solutions to MECD problem are still slightly better than the solutions to MWCD problem. Therefore, the simulation results agree on our analysis that the expected contamination degree minimization problem is more suitable to be the model for inhibiting worm spreading.

6.6.2 Shortest Step Path Blocking Probability

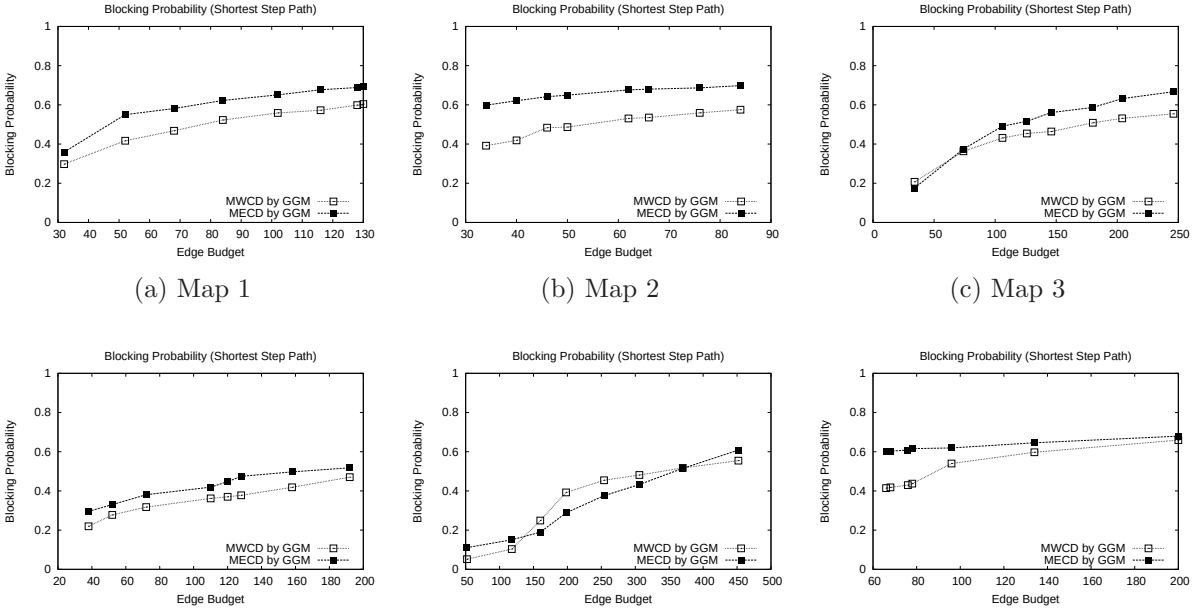
The blocking probability shows how likely a worm propagating along road network would be sniffed by an MVSU. The more MVSUs are deployed, the higher probability a worm would be blocked by an MVSU. The blocking probability also shows how likely a susceptible vehicle would receive the update of the blacklist while moving on the road networks. A vehicle will have more chance to receive the update if more MVSUs could be deployed. So the blocking probability should increase along with the rising edge budget. For the same



(a) Map 1 (b) Map 2 (c) Map 3

(d) Map 4 (e) Map 5 (f) Map 6

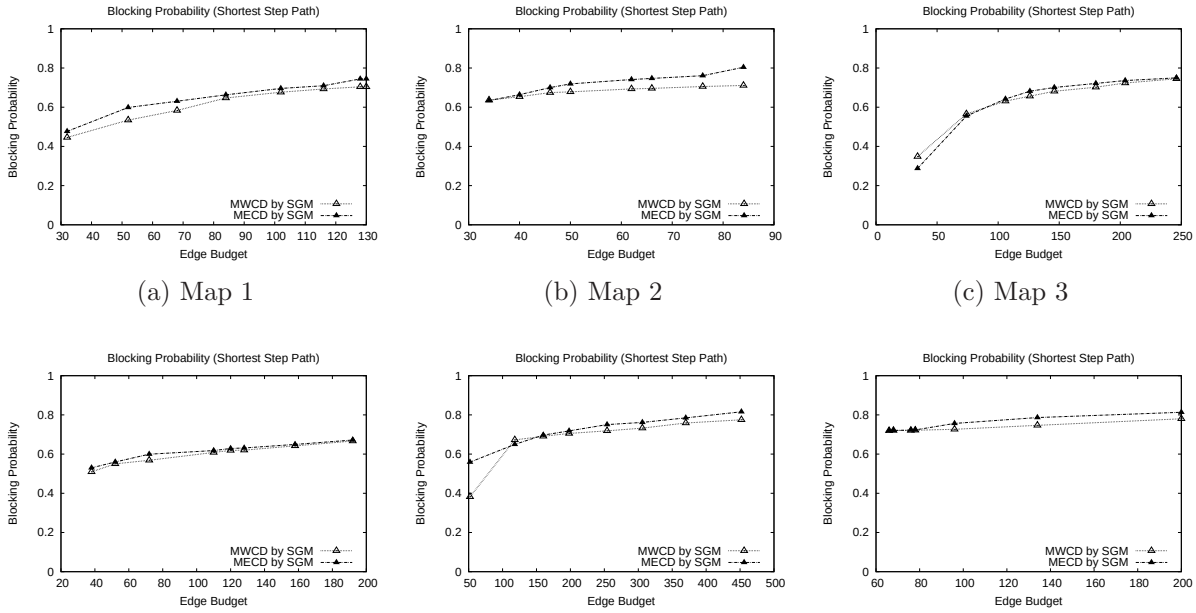
Figure 6.11: Shortest Step Path Blocking Probability (Solutions from OGM)



(a) Map 1 (b) Map 2 (c) Map 3

(d) Map 4 (e) Map 5 (f) Map 6

Figure 6.12: Shortest Step Path Blocking Probability (Solutions from GGM)



(a) Map 1 (b) Map 2 (c) Map 3

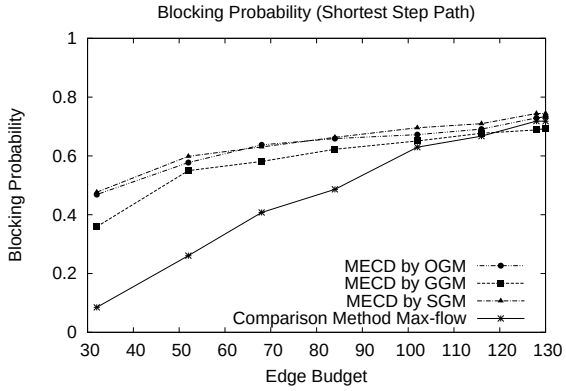
(d) Map 4 (e) Map 5 (f) Map 6

Figure 6.13: Shortest Step Path Blocking Probability (Solutions from SGM)

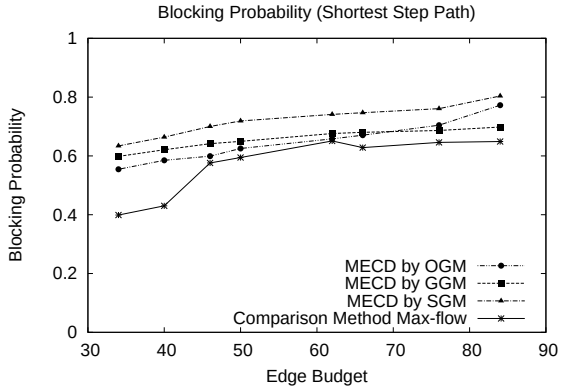
edge budget, the solution leading to the higher blocking probability is considered as the better one.

In this section, the max-flow based method is used as the baseline method. The solutions to MECD found by the three proposed methods are compared to the edge set fond by baseline method. The edge numbers of the edge cuts found by the max-flow based method are used as the edge budget K_{eb} (shown in Table 6.3). For a given edge budget K_{eb}, the edge sets to the minimum expected contamination degree problem found by the three proposed methods are assessed by comparing their blocking probability to the edges sets found by the max-flow method.

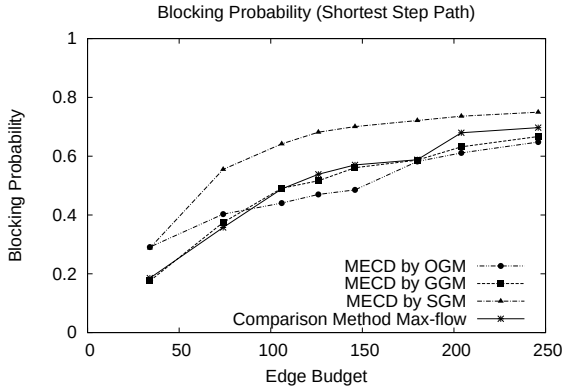
The simulation results of shortest step path blocking probability are shown in Figure 6.14. The solutions found by the three proposed methods show a greater blocking probability on map 1 and 2. In map 3, the solutions from the Original Greedy method and Grid-shrinking Greedy Method are competitive to the solution from the baseline method. The solution from the Simplified Greedy Method is insistently better than the solutions from all other methods. In map 4 and 6, the solutions found by the Original Greedy and Grid-shrinking Greedy Methods outperform the solution from baseline method when the edge budget is relatively small, but the solution from Simplified Greedy Method still surpasses all other methods. In map 5, the solutions from Original Greedy and Simplified Greedy Method also show a greater blocking probability than the baseline method.



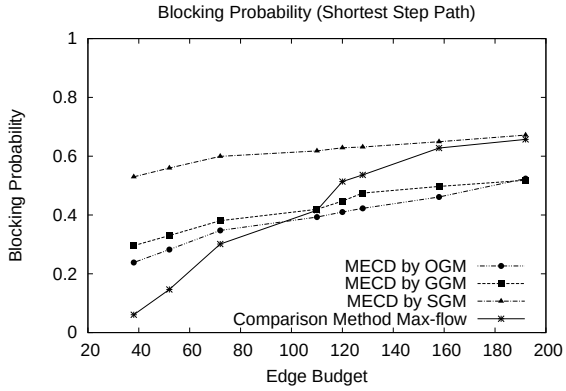
(a) Map 1



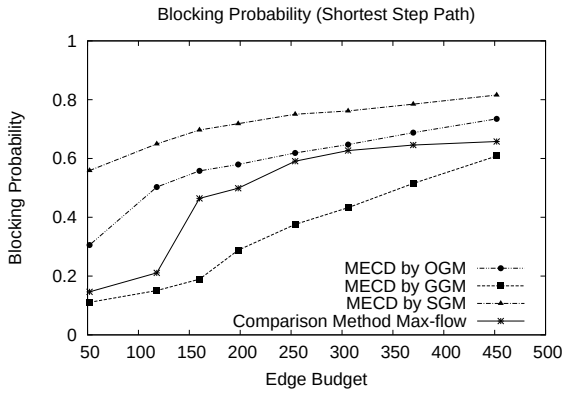
(b) Map 2



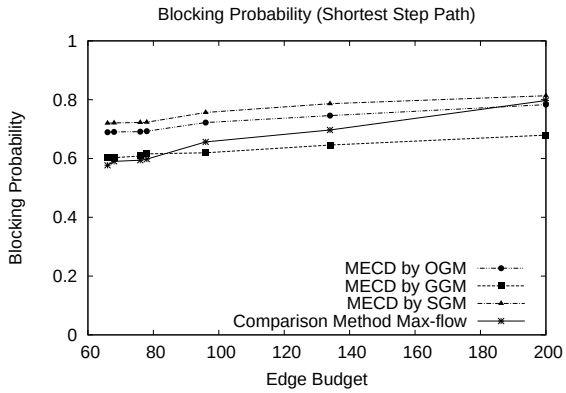
(c) Map 3



(d) Map 4



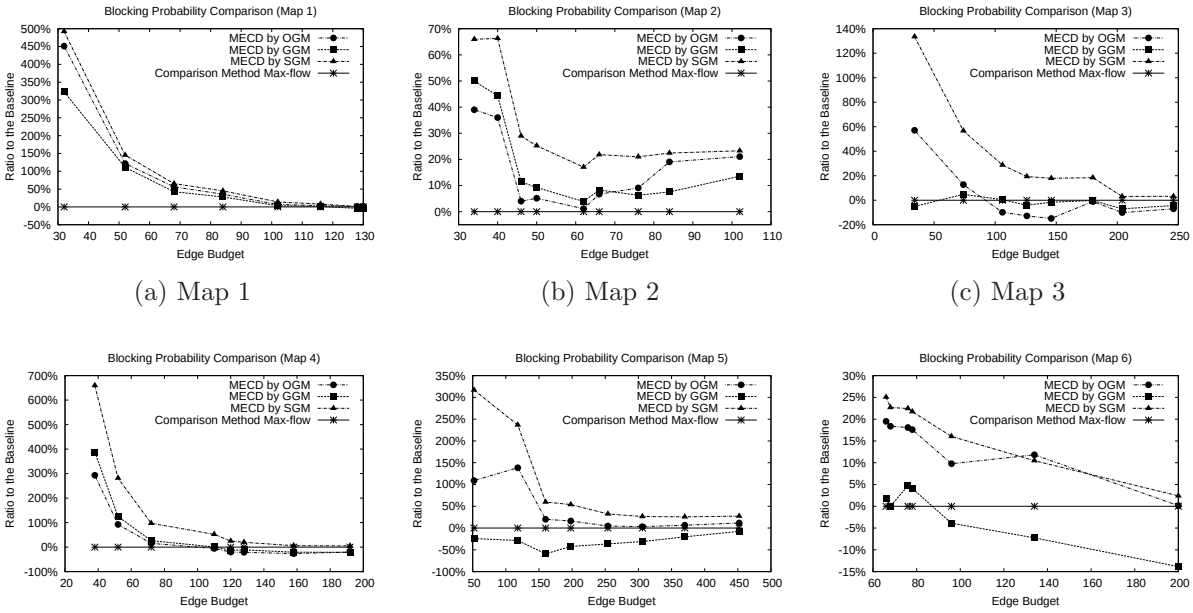
(e) Map 5



(f) Map 6

Figure 6.14: Blocking Probability (MECD on Shortest Step Paths)

However, the blocking probabilities of the solution from Grid-shrinking method are less than the baseline method. It is notable that blocking probability of the solution found by the baseline method is not always increase with the increase of the edge budget. For example, in Figure 6.14b, the blocking probability reaches peak when edge budget is 62 and becomes lower after that. This peculiar behavior is caused by the limitation of the max-flow based method in nature. The max-flow based method always tries to part the graph. The solution quality is independent of the size of the two apart subgraphs. Blocking probability is maximized when the edge-cut equally parts the graph into two subgraphs. Therefore, the blocking probability may drop even with a greater edge budget. Similar situation can be found in Figure 6.14c and Figure 6.14d. In summary, based on the results in Figure 6.14, it can be found that the performance of the Grid-shrinking Greedy Method is dependent of the input graph. The Simplified Greedy Method is the most robust one among all methods, which insistently outperforms the baseline method and the other two proposed methods.



(a) Map 1 (b) Map 2 (c) Map 3

(d) Map 4 (e) Map 5 (f) Map 6

Figure 6.15: Solution Comparison by Blocking Probability

Similarly, we normalize the solution performance by comparing the blocking probability of the solutions from proposed methods to that from the baseline method. The normalized performance curves are shown in 6.15. It can be found that the three proposed methods are significantly superior to the baseline method when the edge budget is relatively small for map 1, 2 and 4. In map 3, the blocking probabilities of the solution found by Original Greedy and Grid-shrinking Greedy Methods are comparable to the baseline method. In

map 5 and 6, the solutions from the Original Greedy and Simplified Greedy Method are insistently superior to the baseline method for all edge budgets, but the solution from Grid-shrinking Greedy Method is worse than the solution of baseline method. The solutions from Simplified Greedy Method surpass other methods in general for all six input maps.

In summary, the performance of Original Greedy and Simplified Greedy Method are robust on almost all the six maps, but the Grid-shrinking Greedy Method only exceeds the baseline method in map 1, 2 and 4. The simulation results presented by blocking probability also agree on the conclusions from Section 6.3 and Section 6.5. The Simplified Greedy Method is reliable and superior to other proposed methods and the baseline method. The application of Grid-shrinking Greedy Method is limited by the road topology due to the potential topology altering issue caused by grid shrinking.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

We addressed the worm containment problem on VANETs. Instead of modeling the worm propagation on the VANET topology, the underlying road network was employed to study the worm spreading and the countermeasure. The worm spreading by V2V multi-hop communication and physical carrying-spreading were combined and modeled as a diffusion process starting from the initial location of the infectious vehicular node. The MVSU deployment problem was proposed for worm spreading containment on VANETs. We showed that the objective of MVSU deployment problem was the same as minimum contamination problem and proposed the expected contamination minimization problem.

The Original Greedy method, Grid-shrinking Greedy Method, and Simplified Greedy Method were developed to solve the minimum expected contamination problem on road networks. The performance of the three proposed methods was evaluated by simulations on real road networks and compared with the classic max-flow based method. The evaluations were carried out by expected and worst contamination degree and average blocking probability. We conclude that the Original Greedy method developed for the social network can be applied to road network topologies, but the complexity is the highest among the three proposed methods. Compared with the Original Greedy method, the Grid-shrinking Greedy Method incorporates the characteristics of road networks. Although the Grid-shrinking Greedy Method is limited by topology characteristics, for some road networks, it could find a better solution with lower complexity. The Simplified Greedy Method overcomes the limitations of Grid-shrinking Greedy Method and outperforms both the Original Greedy method and Grid-shrinking Greedy Method from solution quality and complexity aspects. According to the analysis and simulation results, the proposed methods outper-

form the comparison methods in the following perspectives: The proposed methods are capable of finding a solution for any input K_{eb} with a one-time run. The blocking probability of the solution from the proposed methods is greater than the comparison method in general. The advantages of the proposed methods are remarkable regarding blocking probability when K_{eb} is relatively small.

7.2 Future Work

In this thesis, we explored the application of contamination control model for studying the worm spreading on VANETs. There are many research interests could be further investigated, which are summarized as following directions.

Percolation Probability In our search, the percolation threshold was used as the uniform percolation probability. However, the percolation probability could be vary on individual edge. The statistic data of traffic follow can be employed to estimate the percolation probability of each road section. From a V2V communication point of view, the percolation probability can be found by the traffic density, vehicle velocity, and communication factors. Regarding the vehicle mobility, the statistic data of the origination and destination of vehicles can be used to find the percolation probability to characterize the traffic follow.

Blocking Strategy Our study focused on the edge blocking strategy due to the concerns of the processing capability of infrastructures. However, a single infrastructure could cover more road sections when it is deployed at an intersection. It is worth to study the node blocking or mixed blocking strategy to control the worm spreading on a road network.

Optimal Location of MVSUs In our study, the MVSU was assumed to be a short range communication unit. When the infrastructure communication range increases, the optimal locations of MVSUs are doubtful. In that case, the location of the MVSUs may not be the same as the road sections found by solving contamination minimization problem. More study is deserved when the effective communication range of infrastructure is considered.

Special Pattern Clustering The performance of the proposed Grid-shrinking Greedy Method is dependent of the topology of the input graph. It is worth to explore the clustering method that can keep the main topology characteristics of the input graph.

The vehicular nodes clustering approaches could also be the potential techniques to study the worm inhibiting methods for VANETs [99] [100].

Minimization Contamination Problem The approach of solving contamination minimization problem on the large-scale networks is limited due to the complexity. The greedy method with percolation approximation is the most practical approach for large-scale networks. More study is expected to explore other methods to overcome the complexity issue during contamination degree estimation stage.

References

[1] Nagwan Abdelsamee. Investigating different factors affecting the infection rate of internet worms. pages 1–4, 2016.

[2] Kaouther Abrougui, Azzedine Boukerche, and Richard Werner Nelem Pazzi. Design and evaluation of context-aware and location-based service discovery protocols for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):717–735, 2011.

[3] Osama Abumansoor and Azzedine Boukerche. A secure cooperative approach for nonline-of-sight location verification in vanet. *IEEE Transactions on Vehicular Technology*, 61(1):275–285, 2012.

[4] D Adu-Gyamfi, Yong Wang, Fengli Zhang, M Kamenyi Domenic, Imran Memon, and Yankson H Gustav. Modeling the spreading behavior of passive worms in mobile social networks. 1:380–383, 2013.

[5] Shi An, Jianxun Cui, Meng Zhao, and Jian Wang. A study of network violator interception based on a reliable game model. *Mathematical Problems in Engineering*, 2014, 2014.

[6] Athanasios Bamis, Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks. *Computer Networks*, 52(1):130–154, 2008.

[7] Robert L Barnett, D Sean Bovey, Robert J Atwell, and Lowell Bruce Anderson. Application of the maximum flow problem to sensor placement on urban road networks for homeland security. *Homeland Security Affairs*, 3(3), 2007.

[8] Pavlos Basaras, Ioannis Belikaidis, Leandros Maglaras, and Dimitrios Katsaros. Blocking epidemic propagation in vehicular networks. pages 1–8, 2016.

[9] Rodolfo Bezerra Batista, Azzedine Boukerche, and Alba Cristina Magalhaes Alves de Melo. A parallel strategy for biological sequence alignment in restricted memory space. *Journal of Parallel and Distributed Computing*, 68(4):548–561, 2008.

[10] A. Boukerche. Algorithms and protocols for wireless sensor networks, 2008.

[11] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *computer communications*, 28(10):1193–1203, 2005.

[12] Azzedine Boukerche and Kaouther Abrougui. An efficient leader election protocol for mobile networks. pages 1129–1134, 2006.

[13] Azzedine Boukerche and Luciano Bononi. Simulation and modeling of wireless, mobile, and ad hoc networks. *Mobile ad hoc networking*, pages 373–409, 2004.

[14] Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. *Computer communications*, 29(4):477–489, 2006.

[15] Azzedine Boukerche, Xuzhen Cheng, and Joseph Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.

[16] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Analysis of a randomized congestion control scheme with dsdv routing in ad hoc wireless networks. *Journal of Parallel and Distributed Computing*, 61(7):967–995, 2001.

[17] Azzedine Boukerche, Sajal K Das, Alessandro Fabbri, and Oktay Yildiz. Exploiting model independence for parallel pcs network simulation. In *Parallel and Distributed Simulation, 1999. Proceedings. Thirteenth Workshop on*, pages 166–173. IEEE, 1999.

[18] Azzedine Boukerche and Caron Dzermajko. Performance evaluation of data distribution management strategies. *Concurrency and Computation: Practice and Experience*, 16(15):1545–1573, 2004.

[19] Azzedine Boukerche, Khalil El-Khatib, Yuefei Xu, and Larry Korba. A novel solution for achieving anonymity in wireless ad hoc routing protocol. 2004.

[20] Azzedine Boukerche, Xin Fei, and Regina B Araujo. An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange. *Computer Communications*, 30(14):2708–2720, 2007.

[21] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. A distributed algorithm for dynamic channel allocation. *Mobile Networks and Applications*, 7(2):115–126, 2002.

[22] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. An efficient synchronization scheme of multimedia streams in wireless and mobile systems. *IEEE transactions on Parallel and Distributed Systems*, 13(9):911–923, 2002.

[23] Azzedine Boukerche and Xu Li. An agent-based trust and reputation management scheme for wireless sensor networks. *GLOBECOM'05. IEEE Global Telecommunications Conference, 2005.*, 3:5–pp, 2005.

[24] Azzedine Boukerche, Renato B Machado, Kathia RL Jucá, João Bosco M Sobral, and Mirela SMA Notare. An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Computer Communications*, 30(13):2649–2660, 2007.

[25] Azzedine Boukerche, Anahit Martirosyan, and Richard Pazzi. An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks. *Mobile Networks and Applications*, 13(6):614–626, 2008.

[26] Azzedine Boukerche and Sotiris Nikoletseas. Protocols for data propagation in wireless sensor networks. In *Wireless Communications Systems and Networks*, pages 23–51. Springer, 2004.

[27] Azzedine Boukerche and Mirela Sechi M Annoni Notare. Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing*, 62(9):1476–1490, 2002.

[28] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo Freire Nakamura, and Antonio AF Loureiro. Dv-loc: a scalable localization protocol using voronoi diagrams for wireless sensor networks. *IEEE Wireless Communications*, 16(2):50–55, 2009.

[29] Azzedine Boukerche and Richard Werner Nelem Pazzi. Remote rendering and streaming of progressive panoramas for mobile devices. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 691–694. ACM, 2006.

[30] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina B Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol. In *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l*, pages 560–567. IEEE, 2005.

[31] Azzedine Boukerche, Richard WN Pazzi, and Jing Feng. An end-to-end virtual environment streaming technique for thin mobile devices over heterogeneous networks. *Computer Communications*, 31(11):2716–2725, 2008.

[32] Azzedine Boukerche and Yonglin Ren. A security management scheme using a novel computational reputation model for wireless and mobile ad hoc networks. pages 88–95, 2008.

[33] Azzedine Boukerche and Yonglin Ren. A trust-based security system for ubiquitous and pervasive computing environments. *Computer Communications*, 31(18):4343–4351, 2008.

[34] Azzedine Boukerche and Yonglin Ren. A secure mobile healthcare system using trust-based multicast scheme. *IEEE Journal on Selected Areas in Communications*, 27(4):387–399, 2009.

[35] Azzedine Boukerche, Cristiano Rezende, and Richard W Pazzi. Improving neighbor localization in vehicular ad hoc networks to avoid overhead from periodic messages. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE, 2009.

[36] Azzedine Boukerche and Steve Rogers. Gps query optimization in mobile and wireless networks. pages 198–203, 2001.

[37] Azzedine Boukerche and Amber Roy. Dynamic grid-based approach to data distribution management. *Journal of Parallel and Distributed Computing*, 62(3):366–392, 2002.

[38] Azzedine Boukerche and Samer Samarah. A novel algorithm for mining association rules in wireless ad hoc sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(7):865–877, 2008.

[39] Azzedine Boukerche and Carl Tropper. A static partitioning and mapping algorithm for conservative parallel simulations. In *ACM PADS and SIGSIM Simulation Digest*, volume 24, pages 164–172. ACM, 1994.

[40] Azzedine Boukerche and Carl Tropper. A distributed graph algorithm for the detection of local cycles and knots. *IEEE Transactions on Parallel and Distributed Systems*, 9(8):748–757, 1998.

[41] Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z Ahmad, Ladislau Bölöni, and Damla Turgut. Routing protocols in ad hoc networks: A survey. *Computer networks*, 55(13):3032–3080, 2011.

[42] Simon R Broadbent and John M Hammersley. Percolation processes. 53(03):629–641, 1957.

[43] Yuriy Bulygin. Epidemics of mobile worms. pages 475–478, 2007.

[44] Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468, 2000.

[45] Lin Cheng, Benjamin E Henty, Daniel D Stancil, Fan Bai, and Priyantha Mudalige. Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 ghz dedicated short range communication (dsrc) frequency band. *IEEE Journal on Selected Areas in Communications*, 25(8):1501–1516, 2007.

[46] Lin Cheng and Rahul Shakya. Worm spreading and patching in inter-vehicle communications. *International Journal of Communication Networks and Information Security*, 2(1):50, 2010.

[47] OpenStreetMap contributors. Openstreetmap, 2015.

[48] OpenStreetMap contributors. Planet dump [data file from 2015-11-15]. retrieved from http://www.openstreetmap.org/export♯map=13/40.6350/-73.9699 layers=t, 2015.

[49] OpenStreetMap contributors. Planet dump [data file from 2015-11-15]. retrieved from http://www.openstreetmap.org/export♯map=13/40.7153/-73.9500 layers=t, 2015.

[50] OpenStreetMap contributors. Planet dump [data file from 2015-11-15]. retrieved from http://www.openstreetmap.org/export♯map=13/45.2890/-75.8885 layers=t, 2015.

[51] OpenStreetMap contributors. Planet dump [data file from 2015-11-15]. retrieved from http://www.openstreetmap.org/export♯map=14/43.6504/-79.4498 layers=t, 2015.

[52] OpenStreetMap contributors. Planet dump [data file from 2015-11-15]. retrieved from http://www.openstreetmap.org/export♯map=14/43.7050/-79.4198 layers=t, 2015.

[53] OpenStreetMap contributors. Planet dump [data file from 2015-11-15]. retrieved from http://www.openstreetmap.org/export♯map=14/45.4214/-75.6989 layers=t, 2015.

[54] Felipe Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Viana, Raquel AF Mini, and Antonio AF Loureiro. Data communication in vanets: Protocols, applications and challenges. *Ad Hoc Networks*, 44:90–103, 2016.

[55] NetworkX Developers. Networkx, 2016.

[56] Elie El Ajaltouni, Azzedine Boukerche, and Ming Zhang. An efficient dynamic load balancing scheme for distributed simulations on a grid infrastructure. In *Distributed Simulation and Real-Time Applications, 2008. DS-RT 2008. 12th IEEE/ACM International Symposium on*, pages 61–68. IEEE, 2008.

[57] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Performance analysis of a distributed comparison-based self-diagnosis protocol for wireless ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pages 165–172. ACM, 2006.

[58] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. A distributed fault identification protocol for wireless and mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 68(3):321–335, 2008.

[59] Claude Fachkha and Mourad Debbabi. Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization. *IEEE Communications Surveys & Tutorials*, 18(2):1197–1227, 2016.

[60] Su Fei, Lin Zhaowen, and Ma Yan. A survey of internet worm propagation models. pages 453–457, 2009.

[61] Andy Greenberg. Hackers Remotely Kill a Jeep on the Highway With Me in It, 2015.

[62] Bo Gu, Xiaoyan Hong, and Pu Wang. Modeling worm propagation through hidden wireless connections. pages 1–7, 2009.

[63] Ajay Gupta and Daniel C DuVarney. Using predators to combat worms and viruses: A simulation-based study. pages 116–125, 2004.

[64] Hannes Hartenstein, Bernd Bochow, André Ebner, Matthias Lott, Markus Radimirsch, and Dieter Vollmer. Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project. pages 259–262, 2001.

[65] F Heinzle, KH Anders, and M Sester. Automatic detection of pattern in road networks-methods and evaluation. 36:4, 2007.

[66] Baik Hoh and Marco Gruteser. Computer ecology: Responding to mobile worms with location-based quarantine boundaries. In *Mobile and Wireless Network Security and Privacy*, pages 143–166. Springer, 2007.

[67] Stamatis Karnouskos. Stuxnet worm impact on industrial cyber-physical system security. pages 4490–4494, 2011.

[68] Syed A Khayam and Hayder Radha. Analyzing the spread of active worms over vanet. pages 86–87, 2004.

[69] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Minimizing the spread of contamination by blocking links in a network. 8:1175–1180, 2008.

[70] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Solving the contamination minimization problem on networks for the linear threshold model. pages 977–984, 2008.

[71] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(2):9, 2009.

[72] Masahiro Kimura, Kazumi Saito, and Ryohei Nakano. Extracting influential nodes for information diffusion on a social network. 7:1371–1376, 2007.

[73] Neal Leavitt. Mobile phones: the next frontier for hackers? *IEEE Computer*, 38(4):20–23, 2005.

[74] Angsheng Li and Linqing Tang. The complexity and approximability of minimum contamination problems. pages 298–307, 2011.

[75] Pele Li, Mehdi Salour, and Xiao Su. A survey of internet worm detection and containment. *IEEE Communications Surveys & Tutorials*, 10(1):20–35, 2008.

[76] Zhaowen Lin, Fei Su, and Yan Ma. The analysis of internet worm modeling in ipv4 and ipv6 networks. pages 738–743, 2010.

[77] Rohas Nagpal. Cyber terrorism in the context of globalization. pages 1–23, 2002.

[78] Maziar Nekovee. Modeling the spread of worm epidemics in vehicular ad hoc networks. 2:841–845, 2006.

[79] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[80] Paul F. Roberts. Blaster worm attack a bust, 2003.

[81] Yonglin Ren and Azzedine Boukerche. Modeling and managing the trust for wireless and mobile ad hoc networks. pages 2129–2133, 2008.

[82] Yaman Roumani, Joseph K Nwankpa, and Yazan F Roumani. Examining the relationship between firms financial records and security vulnerabilities. *International Journal of Information Management*, 36(6):987–994, 2016.

[83] Samer Samarah, Muhannad Al-Hajri, and Azzedine Boukerche. A predictive energy-efficient technique to support object-tracking sensor networks. *IEEE Transactions on Vehicular Technology*, 60(2):656–663, 2011.

[84] David E. Sanger. Obama order sped up wave of cyberattacks against iran, 2012.

[85] Sandeep Sarat and Andreas Terzis. On using mobility to propagate malware. pages 1–8, 2007.

[86] John Tang, Cecilia Mascolo, Mirco Musolesi, and Vito Latora. Exploiting temporal complex network metrics in mobile malware containment. pages 1–9, 2011.

[87] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

[88] Oscar Trullols-Cruces, Marco Fiore, and Jose M Barcelo-Ordinas. Understanding, modeling and taming mobile malware epidemics in a large-scale vehicular network. pages 1–9, 2013.

[89] Oscar Trullols-Cruces, Marco Fiore, and Jose M Barcelo-Ordinas. Worm epidemics in vehicular networks. *IEEE Transactions on Mobile Computing*, 14(10):2173–2187, 2015.

[90] Leandro A Villas, Azzedine Boukerche, Horacio ABF De Oliveira, Regina B De Araujo, and Antonio AF Loureiro. A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks. *Ad Hoc Networks*, 12:69–85, 2014.

[91] Jian Wang, Yanheng Liu, and Kevin Deng. Modelling and simulating worm propagation in static and dynamic traffic. *IET Intelligent Transport Systems*, 8(2):155–163, 2014.

[92] Yini Wang, Sheng Wen, Yang Xiang, and Wanlei Zhou. Modeling the propagation of worms in networks: A survey. *IEEE Communications Surveys & Tutorials*, 16(2):942–960, 2014.

[93] Daniel M Watkins, Leticia Cuéllar, Deborah A Kubicek, Erick Rodriguez, and Phillip D Stroud. Identifying security checkpoint locations to protect the major us urban areas. *Homeland Security Affairs*, 11, 2015.

[94] Nicholas Weaver, Vern Paxson, Stuart Staniford, and Robert Cunningham. A taxonomy of computer worms. pages 11–18, 2003.

[95] Nawaporn Wisitpongphan, Fan Bai, Priyantha Mudalige, Varsha Sadekar, and Ozan Tonguz. Routing in sparse vehicular ad hoc wireless networks. *IEEE journal on Selected Areas in Communications*, 25(8):1538–1556, 2007.

[96] Evsen Yanmaz. Epidemic propagation in overlaid wireless networks. pages 1–5, 2008.

[97] Jin Y Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.

[98] Lidong Zhai, Wei Guo, Zhaopeng Jia, Li Guo, and Jinqiao Shi. Worm propagation model for heterogeneous network. pages 151–154, 2012.

[99] Zhenxia Zhang, Azzedine Boukerche, and Richard Pazzi. A novel multi-hop clustering scheme for vehicular ad-hoc networks. pages 19–26, 2011.

[100] Zhenxia Zhang, Richard W Pazzi, and Azzedine Boukerche. A mobility management scheme for wireless mesh networks based on a hybrid routing protocol. *Computer Networks*, 54(4):558–572, 2010.

[101] Qinhua Zheng, Ting Liu, Xiaohong Guan, Yu Qu, and Na Wang. A new worm exploiting ipv4-ipv6 dual-stack networks. pages 9–15, 2007.

[102] Cliff C Zou, Don Towsley, and Weibo Gong. On the performance of internet worm scanning strategies. *Performance Evaluation*, 63(7):700–723, 2006.

[103] M Zulkiflee, SA Azirah, N Haniza, A Zakiah, and S Shahrin. Behavioral analysis on ipv4 malware on different platforms in ipv6 network environment. pages 74–79, 2011.