UNIVERSITY OF CINCINNATI

Date: January 4, 2005

I, Xuguang Wang,

hereby submit this work as part of the requirements for the degree of:

Doctorate of Philosophy (Ph.D.)

in:

Geography

It is entitled:

Spatial Adaptive Crime Event Simulation with
the RA/CA/ABM Computational Laboratory

This work and its defense approved by:

Chair: Dr. Lin Liu

Dr. Howard Stafford

Dr. Tony Grubesic

Dr. John Eck

Spatial Adaptive Crime Event Simulation with the RA/CA/ABM Computational Laboratory

A dissertation submitted to the
Division of Research and Advanced Studies
of the University of Cincinnati

in partial fulfillment of the
requirements for the degree of

DOCTORATE OF PHILOSOPHY (Ph.D.)

in the Department of Geography
of the College of Arts and Sciences

2005

by

Xuguang Wang

B.S., Peking University, 1997
M.S., Chinese Academy of Surveying and Mapping, 2000

Committee Chair: Dr. Lin Liu

Abstract

An agent-based crime event and crime pattern simulation model is developed in this research. The purpose of the simulation model is to provide a computational laboratory for environmental criminologists to study the interactions among offenders, targets, controllers, and crime places. The simulation model also aims to provide a useful tool for teaching crime event theories.

Routine activity theory and crime pattern theory are the theoretical foundations of the simulation model. Agent-based modeling coupled with cellular automata addresses the complex crime event process of street robbery. A type of spatial autonomous agent is developed with a wayfinding capability on the urban street network. The wayfinding algorithm is based on a reinforcement learning algorithm. Offender agents, target agents and police agents are developed based on the spatial autonomous agent, which can be released on a street network to execute their routine activity schedules. The interactions among offender agents, target agents, police agents, and crime places create crime events and crime patterns for analysis. Offender agents and target agents can learn from their past offending/victimization experience and change their spatial behaviors.

The crime event and crime pattern simulation model is tested to be able to generate credible spatial, temporal, victimization, and offending patterns. The simulation model is then applied to examine the effect of agent adaptations on spatial crime patterns, offending patterns and victimization patterns. The power-function distributions of crime events among crime places and offender population are examined as emphases.

Targeted for MS Windows desktop, the RA/CA/ABM computational laboratory is implemented using Visual C++. The computational laboratory has a graphic user interface that allows users to customize the simulation model, control the simulation process, visualize agent movement and crime patterns during the simulation, and query agent properties and crime patterns during the simulation. The computational laboratory is loosely coupled with Arcview GIS, so that it can take spatial data from Arcview as input, and make use of spatial analytical capabilities of Arcview to analyze the output crime patterns.

Keywords: agent-based modeling, cellular automata, crime pattern simulation, Geographic Information Systems, routine activity theory

Acknowledgement

This research would not be possible without the support of many individuals and research grants. First of all, I would like to express my deepest appreciation to my advisor and committee chair, Dr. Lin Liu, for guiding me through the completion of this dissertation. Dr. Liu has spent countless hours in discussing with me about the research, and he has given me many valuable advices throughout this process. His advices assure the right direction of this research, and contribute significantly to the positive outcome of this research.

I am grateful to my committee members Drs. Howard Stafford, Tony Grubesic, and John Eck. Dr. Eck introduced me to the literature of environmental criminology, and spent many hours to talk with me about this research. His enthusiasm and interest in this research impressed me and helped me a lot. Dr. Stafford and Dr. Grubesic gave me valuable feedback to both my proposal and dissertation.

I also would like to thank other professors, staff, and fellow graduate students in the department of geography for their help during this process. Special thanks are extended to Dr. Robert South, Dr. Robert Frohn, and Dr. Kenneth M. Hinkel for giving me a lot of encouragement in finishing this dissertation; to Nuha Nasrallah for administrative assistance; and to Mark deSocio, Johnathan Baker, and Ezgi Akpinar for reviewing my draft.

This research would not be possible without the love and support of my wife, my son, my parents, brothers and sister. During these years, my wife is very patient and helpful for my study. She took care of our son and household work while I spent most of the time in the lab. I owe a great debt to her. This dissertation is dedicated to my family.

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0081434. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

List of figures and tables

Tables

Chapter 1 Introduction

1.1 Introduction

This research aims to design, implement, and test an agent-based computational laboratory for environmental criminologists to study interactions among offenders, targets, controllers and crime places. Agent-based modeling is a new simulation method used by geographers to model human-environment interactions. For example, Gimblett, et. al. (2002) created an agent-based model for recreation behavior simulation in Grand Canyon National Park, and Dibble (2001) created computational laboratories for simulating globalization process in spatial networks.

Environmental criminology is the subject that studies the spatial dimension of crime (Brantingham & Brantingham, 1991). Traditionally, researchers have used two approaches to study spatial patterns of crime. They develop theories of crime that specify conditions and factors leading to the occurrence of crime and formation of crime patterns. They also examine empirical crime patterns, either to reveal common features or to test theories. Existing crime event theories, such as routine activity theory (Cohen & Felson, 1979), are static. Routine activity theory focuses on local site features and situations that lead to a crime event. The likelihood of a crime event at a certain place is a balance between the statuses of the crime event components, as expressed in a formula by Eck (1995). However, the theory does not address the feedback effects of crime events on local situations. According to the feedback hypothesis by Eck (2003) and the complex crime event process summarized by crime pattern theory (Brantingham & Brantingham, 1993), feedback exists between crime events and local situations. Therefore, the system of crime event components is a complex adaptive system. According to complexity theory, the behavior of the system is difficult to understand using a regular modeling approach.

Simulation approach for crime pattern study is a new research direction suggested by several environmental criminologists (Eck, 1995; Brantingham & Brantingham, 2003). *SPatial Adaptive Crime Event Simulation* (SPACES) is a formal definition of this new research direction with collaboration by the Department of Geography and Criminal Justice Division at the University of Cincinnati. The RA/CA/ABM computational laboratory developed in this dissertation research is a prototype implementation of SPACES.

A simulation approach to study spatial crime patterns combines the advantages of the two traditional approaches, and overcomes some problems associated with them. Compared with the static crime event theories, simulation can capture the complex crime event process. For example, in the simulation model developed in this study, researchers can incorporate interaction rules and test the implications of these rules on crime patterns. Thus, by making crime theories dynamic, it is easier to reveal their implications for crime patterns. Furthermore, a simulation model generates hypothetical crime patterns which can be compared to actual patterns, thereby allowing fuller exploration of theories.

The RA/CA/ABM computational laboratory introduced in this dissertation provides an agent-based crime pattern simulation model derived from routine activity, and provides a Windows program to wrap the model and allow environmental criminologists to customize the model to make their own simulation experiments. Routine activity theory (Cohen & Felson, 1979) argues that for a crime event to occur, there must be a motivated offender and a desirable target present at a certain place, and guardians must be absent. Therefore, the agent-based crime pattern simulation model developed in the computational laboratory consists of four types of agents:

offender agents, target agents, police agents, and place agents. A type of spatial autonomous agent is developed as the base class for offender agents, target agents, and police agents. Being autonomous is in the sense that such agents are able to execute designed routine activity schedules in space and time without external intervention. Crime event rules are developed based on the crime event likelihood evaluation formula developed by Eck (1995). A typical simulation experiment with the computational laboratory involves a number of offender agents, target agents, and police agents released on a street network to perform some routine activities. Their interactions create crime events and crime patterns, which are recorded in log files for analysis.

The RA/CA/ABM computational laboratory is based on the commercial robbery simulation model developed by Liang, et. al. (2001) and the street robbery simulation model (the RA/CA model) developed by Liu, et. al. (2004). The limitation of these two models is that agent spatial behaviors are not adaptable. This research improves agent spatial behavior simulation using artificial intelligence, and agent spatial behaviors become adaptable.

Street robbery is still the crime type addressed by the RA/CA/ABM computational laboratory. Robbery is defined as "theft and attempted theft, directly from a person or commercial establishment, of property or cash by force or threat of force, with or without a weapon" (Ward and Ward, 1975). Street robbery is a type of robbery whose targets are pedestrians.

1.2 Challenges facing crime places research

Environmental criminologists have been interested in crime places since the 1970s. Now crime place has become an important perspective in crime event theories and crime prevention practice. Routine activity theory justifies the importance of crime places in crime theories and crime prevention (Eck, 1995), which argues that for a crime event to occur, an offender and a target must converge at the same place with guardians absent from that place.

In contrast with neighborhood analysis, the focus on crime places defined a new spatial scale for crime analysis. Eck (1995) defined a crime place as "a very small area, usually a street corner, address, building, or street segment". However, this shift of analysis unit to crime places brings several challenges to environmental criminology, challenges that a computational laboratory can fit in. These challenges are discussed as follows.

A. Testing routine activity theory at micro crime place level

In contrast with neighborhood crime pattern studies (which are usually from an offender motivation perspective), examining crime patterns at place level often involves a crime event perspective (Eck, 1995). Routine activity theory is a crime event theory in explaining crime patterns among micro crime places. To examine the explanation power of routine activity theory on crime patterns at micro place level, one approach is to collect crime event data and routine activity data at a place, and analyze their relationship. This empirical approach is not ideal because data at such detail is difficult to obtain.

An alternative approach is to use computer simulation. Agent-based computational laboratories allow us to model the behavior of individuals and the interaction between individuals. The crime patterns resulting from these interactions can then be observed and compared with actual crime patterns. If we can observe crime patterns from simulations that conform to known crime patterns in the real world, then the explanation power of routine activity theory is tested.

B. Observing global crime patterns as the result of local interactions between components of crime events

The crime event perspective in environmental criminology reflects an idea that macro crime patterns are the result of local interactions between potential offenders, targets, controllers and crime places. For example, Eck (1995) pointed out that crime patterns result from complex space-time interactions among the components of routine activity theory.

It has been established that the distribution of crime rates over space (at various scales) is not random. At the places level, crime rates have a skewed distribution. A few places account for a disproportionately large number of crime events compared to other places. For example, Sherman (1995) reported that in Minneapolis, only 3% of the places produced 50% of the calls for service to which police were dispatched. Spelman (1995) showed that the distribution of crime among crime places is in a power-function form.

The original version of routine activity theory can explain why some places have crime while other places have no crime. However, it could not be directly applied to explain the skewed distribution of crime events among crime places. The agent-based computational laboratory developed in this study sheds light on explaining the skewed distribution of crime patterns, which allows the interaction and feedback rules between offenders, targets, controllers, and environment to be specified in simulation experiments. The simulation experiments show that the skewed distribution of crime events among crime places maybe the result of local interaction among offenders, targets, controllers and environment.

C. Elaborating the relationship among aspects of routine activity theory

According to routine activity theory, potential offenders, potential targets, places, place managers, guardians, and handlers all contribute to the crime rate at for a given place. However, because routine activity theory is static, we don’t know how the change of crime controllers will affect the corresponding edge of the crime triangle. For example, how much change of place management effectiveness is needed before the crime rate at one place is reduced significantly? How much change of desirability of one victim will significantly reduce the chance of repeat victimization of a potential target (similar questions can be found in Sherman, 1995)?

These problems can not be solved using regular mathematical modeling because the interaction among offenders, targets, places, and crime controllers is both nonlinear and includes feedback. Agent-based modeling sheds light on the solving of these problems because we can build artificial agents to represent individuals and specify their interaction rules in a "society" of agents. Then, experiments can be designed and performed in this society, and the interactions among agents and the outcome can be observed and analyzed. Such experiments do not represent any real crimes, but they can inform us about what could happen in the real world.

1.3 Computational laboratory for environmental criminologists

The last section discussed some challenges facing environmental criminological theories that have implications on spatial characteristics of crime. To summarize, the difficulty is caused by the static nature of routine activity theory - a theory that focuses on the local situation and the micro-level interaction at individual places, while it has implications on the global structure of crime patterns. The rules specified by routine activity theory are simple, but there are feedback effects among crime event and crime event components. As Eck (2003) suggested, the system formed by offenders, targets, controllers and their environment is a feedback system. The components of crime events are adaptable. The routine interaction among offenders, targets, and controllers will change the crime patterns; and the changed crime patterns are then fed back which affect future interactions among them. This complex feedback relationship between crime patterns and routine activity patterns is summarized in Figure 1.1.

Given this complex feedback relationship, what will the final crime pattern look like? Will it be a skewed and stable distribution? Or, under this complex environment, will an individual adaptation strategy (self-policing) work in reducing victimization risks? These questions can not be answered by a regular deductive approach because the process is too complex and too many variables and equations will be required. An inductive approach will also face difficulties in answering these questions because the process is hard to observe in the real world.

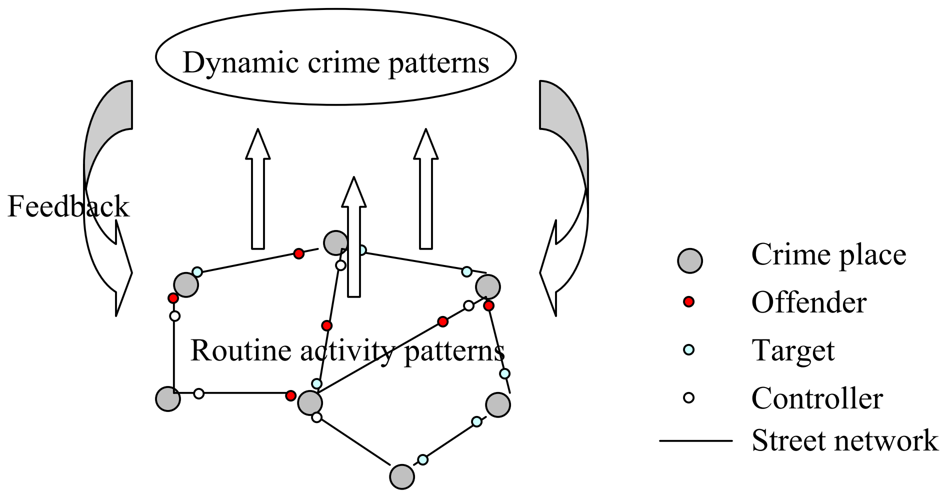


Figure 1.1 The feedback relationship between crime patterns and routine activity patterns

Computational laboratories provide simulation environments for geographers to build theories that involve complex processes. Dibble (2001) formally defined a geographic computational laboratory. The definition that Dibble gave to computational laboratory is “a well specified spatial simulation model coupled with careful experimental design” (Dibble, 2001).

The RA/CA/ABM computational laboratory developed in this dissertation research aims to provide a spatial simulation environment for environmental criminologists to explore the implications of routine activity theory and agent adaptations on crime patterns. The computational laboratory provides an agent-based simulation model of street robbery, and a graphic user interface allowing researchers to design specific experiments according to their needs. Offender agents, target agents and police agents are developed with capabilities in executing routine activity schedules on the street network. Offender agents and target agents are adaptable both in spatial behaviors and non-spatial properties. Therefore, the computational laboratory can be used to model the complex interaction and feedback processes among agents and environment, and test the implications of such feedback on crime patterns.

Previous studies

The idea of crime simulation is not new. Liang (2001) developed a commercial robbery simulation model using cellular automata and routine activity theory. Liu et. al. (2004) developed a street robbery simulation model (RA/CA model) using cellular automata and routine activity theory. The contributions of the two studies are that they put routine activity theory in a spatial simulation environment, and make the theory dynamic. The limitations of these two studies are summarized as follows:

First, in the two studies, routine activities are not explicitly modeled. The activities of agents are modeled by Monte-Carlo method (which is based on a type of mean information field). Offenders' interaction between home and robbery location is controlled by a statistic field (mean information field). Such spatial behavior simulation makes agent spatial learning and adaptation difficult to implement.

Second, agent adaptations in the two studies are non-spatial. Agent spatial adaptations have not been implemented. Following crime events, agents can change their non-spatial properties but can not change their spatial activity pattern. As a result, some spatial patterns of crime could not be simulated by these two models.

Third, guardians are not explicitly modeled as agents, and guardian capability is only a property of target agents. The fact is that other people on the street and police agents protect targets.

This research is based on the two previous studies and overcomes some of their limitations. The major contribution of the RA/CA/ABM computational laboratory is that it addresses the spatial adaptation capabilities for offender agents and target agents. Agent property adaptations remain in the new simulation model. Thus, the new simulation model is able to incorporate agent spatial adaptation rules as well as property adaptation rules into the model. The computational laboratory also provides a better user interface that supports crime simulation experiment design.

In addition to the contribution in crime theories and crime analysis, developing such a computational laboratory represents several challenges in GIS and dynamic space-time process modeling. The RA/CA/ABM computational laboratory develops a type of spatial autonomous agent that can find its way from origin to destination in an urban environment. The computational laboratory also implements a dynamic visualization window for the simulation process.

In summary, this research accomplishes the following:

(1) Develops a type of spatial autonomous agent using a reinforcement learning algorithm that is able to execute routine activity schedules on a street network, and is able to adapt spatial activity patterns according to past experience of crime;

(2) Develops three types of agents for crime simulations (offender agent, target agent, police agent) which inherit from the spatial autonomous agent class;

(3) Specifies a set of crime event rules, adaptation rules and functions for offender agents and target agents;

(4) Develops a general spatial crime simulation model that is based on routine activity theory and spatial autonomous agents, and wraps the simulation model in a Windows graphic user interface to make the RA/CA/ABM computational laboratory;

(5) Designs several spatial-temporal crime pattern simulation experiments to validate the computational laboratory;

(6) Designs several experiments to explore agent adaptations and their effect on spatial crime patterns, offending patterns, and victimization patterns.

1.4 Organization of the dissertation

This dissertation is organized into six chapters. Chapter 1 gave a brief introduction to the RA/CA/ABM computational laboratory; the necessity of building such a computational laboratory; routine activity theory and crime place research; and the major points accomplished in this research.

Chapter 2 is literature review. This chapter presents routine activity theory in more detail, and reviews other related theory. Crime pattern theory (Brantingham & Brantingham, 1993) and the feedback hypothesis by Eck (2003) are introduced. The second review topic is in methodology - cellular automata and agent-based modeling in dynamic space-time process modeling are discussed. This topic covers the definitions of the two modeling methodologies, their current applications in social science in general and in geography specifically, and their integration with geographic information systems. The third topic is the RA/CA crime simulation model which is the basis of this research.

Chapter 3 presents the conceptual components of the RA/CA/ABM computational laboratory. This chapter first presents the base class of spatial autonomous agents, and then presents offender agents, target agents, and police agents for crime simulations, which are derived from the spatial autonomous agent class. Crime places are treated as place agents. Crime event rules and crime templates are then developed for offender agents and target agents. The framework of the spatial crime pattern simulation model is then presented as summary to this chapter.

Chapter 4 discusses issues regarding implementation of the computational laboratory, including the implementation strategy, implementation tool, relationship between the computational laboratory and GIS, logical components of the simulation program, objects model diagram of the program, and user interface.

Chapter 5 presents several carefully designed crime simulation experiments with the RA/CA/ABM computational laboratory. The “timing of crime” and “accessibility and crime” scenarios are to validate the computational laboratory through showing that the simulations can produce spatial-temporal crime patterns conforming to known crime theories and empirical studies. “Repeat location”, “target adaptation and victimization risk”, and “repeat offending” scenarios are designed to explore the effect of agent adaptations on spatial crime patterns, victimization patterns, and offending patterns.

Chapter 6 summaries the RA/CA/ABM computational laboratory, discusses simulation experiment results, contributions of the RA/CA/ABM computational laboratory, and lists limitations and future research directions suggested by this research.

Chapter 2 Literature Review

This chapter reviews the crime theories and assumptions that backup the computational laboratory developed in this research. Also, methodologies used in building the RA/CA/ABM computational laboratory for crime simulation will be reviewed. Finally, previous work in crime event and crime pattern simulation will be reviewed.

2.1 Theoretical Background

Environmental criminology consists of a set of related theories: routine activity theory, crime pattern theory, and rational choice theory. These theories focus on crime events and represent a movement in criminological theories – a deviation from the motivation tradition. This research aims to combine the routine activity approach with an agent-based simulation approach to provide a tool to study agent interactions in crime. These theories are briefly summarized as follows.

2.1.1 Routine activity theory

Basic definition

Routine activity theory is first presented by Cohen & Felson (1979) as an approach to explain the paradoxical crime trends in U.S. During 1960s and 1970s, the general social economic conditions of U.S. were improved, but crime rates increased significantly in the same period for many violent and property crime types.

Routine activity theory argues that for direct contact predatory crime, a crime event is the result of the convergence of three conditions at specific places in space and time - likely offenders, suitable targets, and the absence of capable guardians for the targets. Aided by human ecology theory, Cohen & Felson (1979) argue that the convergence of these three conditions is produced by social structures and routine activities of people. Therefore, the change of social structure and routine activities influence crime rates of the society. Originally, routine activity theory was applied to explain crime trends within highly aggregated spatial areas (for example, US as a whole). More recently, it began to be applied to explain the variation of crime rates at micro-level places.

The original version of routine activity theory contains four components of a crime event - likely offenders, suitable targets, places, and guardians. Later, handlers were added as controllers for offenders (Felson, 1986 in Felson, 1995), and place managers were added as controllers for the behavior in places (Eck, 1994 in Felson, 1995). Thus, with the latest version of routine activity theory, there are six elements to be considered to assess risk of crime at a specific place.

The crime event triangles

Eck (2003) represents the six components of a crime event as two triangles, as shown in Figure 2.1. The edges of inner triangle represent the three basic elements of crime event (offender, target and place); the edges of external triangle represent controllers of crime event (guardian, handler, and place manager). For a crime to occur, the inner elements must be present and the outer elements must be weak or absent. These two triangles give us a direct view of components of crime events and can help us understand routine activity theory.

Figure 2.1 The crime event triangle (Eck, 2003, p89)

Crime event likelihood evaluation and rules for crime event occurrence

Routine activity theory is a micro-level crime event explanation theory (Eck, 1995). It focuses on local situations and interactions among individuals. It tells us that in order to evaluate crime risk at a given place, we need to consider the status of offenders, targets, and controllers simultaneously.

Eck (1995) shows that the internal logic of routine activity theory is that its components are interactive. Linear models could not represent this interactive property. Therefore, Eck (1995) provides a possible formulation of instant crime likelihood evaluation according to routine activity theory:

$$L(S_{tijk}) = \frac{\delta_{tik} T_{tik} \mu_{tijk} O_{tijk} \alpha_{ti} P_{ti}}{(1+\gamma_{tik} G_{tik})(1+\beta_{tijk} H_{tijk})(1+\varepsilon_{ti} M_{ti})} \qquad \ldots\ldots(2.1)$$

$L(S_{tijk})$ is the crime likelihood for situation S_{tijk}. *t* is time, *i* is place, *j* is offender, *k* is offense type (in this study, offense type is street robbery). The variables *T*, *G*, *O*, *H*, *P* and *M* represent respectively the presence of a target, a guardian, an offender, an intimate handler and a place. If either of these elements is present in situation S_{tijk}, the variable takes a value of 1. Otherwise they take a value of 0. δ, γ, μ, β, α and ε represent target desirability, guardian capability, offender motivation, handler intimacy, place accessibility, and management effectiveness. The values of δ, γ, μ, β, α and ε are within the domain of [0, 1].

Equation 2.1 summarizes all conditions specified by routine activity theory for a crime event. For example, if offenders are not present at a place, then *O* equals 0. Thus, *L* equals 0 and there is no possibility for crime. Similarly, if there are no targets, then *T* equals 0, and *L* equals 0. Again, there is no crime. Without a place, *P* equals 0, there is nowhere for offenders and targets to meet, and *L* equals 0, indicating no possibility for crime event. On the other hand, if offenders, targets, and places all present, then the likelihood of crime is determined by the motivation of offenders, desirability of targets, accessibility of places, intimacy of handlers, guardian capability of guardians, and management effectiveness of place managers. The higher *L* is, the higher the possibility for a crime.

Equation 2.1 can easily be transformed into rules for crime event occurrence to be used in computer simulation. For example, the following rules are developed according to equation 2.1:

(1) If there is no offender at a place, then $L = 0$;

(2) If there is no target at a place, then $L = 0$;

(3) If there are offenders and targets at a place, then calculate crime likelihood *L* as follows:

$$L = \frac{\delta_{tik}\mu_{tijk}\alpha_{ti}}{(1+\gamma_{tik}G_{tik})(1+\beta_{tijk}H_{tijk})(1+\varepsilon_{ti}M_{ti})} \quad \ldots\ldots(2.2);$$

(4) Generate a uniform random number *r* in the domain of [0, 1]. If *r* is less than *L*, then a crime event occurs. Otherwise, crime event will not occur.

2.1.2 Crime pattern theory

Crime pattern theory (Brantingham & Brantingham, 1993) takes offender movement into consideration and explains why places in some areas have a greater chance of having offenders than places in other areas. Crime pattern theory argues that the distribution of crime over places is described by the space-time distribution of offenders, targets, and controllers. The distribution of offenders, targets, and controllers over space and time could be predicted by their routine activities.

To summarize, crime pattern theory provides the following points of view that directly inform the construction of the crime simulation model in this research:

1. The space-time distribution and convergence of potential offenders, targets, handlers and guardians will describe crime patterns (Eck & Weisburd, 1995).
2. The space-time distribution of offenders, targets, and controllers are not random, nor is it uniform (Brantingham & Brantingham, 1993). People are on schedules of routine activities, which can predict their spatial position at a specific time.
3. Crime events are rare. Just like ordinary people, potential offenders spend most of their time in legal routine activities.
4. During routine activities, potential offenders develop awareness space about their environment (Brantingham & Brantingham, 1993). The awareness space is where offenders notice and search for criminal targets and places. Most offenders will confine their search area within the areas they became familiar with through routine activities (Eck & Weisburd, 1995). After a certain period of time, the offender will develop a relatively fixed "template" as his offending pattern in awareness space.

5. Criminal activities feed upon legal routine activities, which means that the offence often takes place in an offender's awareness space, as well as that the offence often takes place at the time of legal routine activities.

The above review introduces some basic ideas of routine activity theory and crime pattern theory. In summary, routine activity theory explains crime patterns by what kind of people are present and absent from particular places (Eck, 1995); crime pattern theory explains crime patterns through offender search and learning behavior. Both theories focus on crime events. However, they do not address the dynamics of crime event components over time, and the implications of such dynamics on crime patterns are unclear. This justifies a computational laboratory approach for studying crime patterns, which is able to address the adaptation processes of crime event components.

Routine activity theory and crime pattern theory suggest a framework for spatial adaptive crime event simulation. Since a crime pattern is described by the space-time distribution of offenders, targets, and controllers in their routine activities, it is possible to predict crime patterns through modeling routine activities and offender rational choice within a simulation environment.

2.1.3 Feedback hypothesis

Routine activity theory tends to regard components of crime event as mechanical particles whose convergence in space and time produce crime events and crime patterns. However, human beings are intelligent creatures that can react and adapt according to environment changes. Based on routine activity theory, Eck (2003) develops a feedback view of the human system formed by offenders, targets, and controllers. Routine activity theory explains that the convergence of offenders and targets in space and time generates crime events. However, crime events are still rare. This is because the presence of controllers suppresses many crime events. Therefore, each component of a crime event influences the likelihood of the crime event. On the other hand, Eck argues that the presence of crime event also has influence on offenders, targets, and controllers. For example, the occurrence of crime events at a place may reinforce the motivation of the offender to offend in the future. This may cause a repeat of a crime in a given location. The place manager may react to crime through enforcing management rules, which reduces the future possibility of a crime. This prevents crime from becoming a repeat location problem in a given location. As a result, the number of crime events occurring at a place is decided by a balance of such positive and negative influences. For example, if the place manager fails to respond to the crime problem by improving place management effectiveness, then the place will become a crime hotspot. Such influences can be represented as feedback arrows as shown in Figure 2.2, where the positive sign (+) means "stimulates" or "reinforces" and the negative sign (-) means "suppresses" or "prevents".

For example, Figure 2.2 shows that the involvement of a target in a crime event will stimulate the guardian to increase protection for the target, so that the chance for the target to be involved in crime events is reduced in the future. Therefore, the arrow from event to guardian is annotated by a positive sign, while the arrow from guardian to target is annotated by a negative sign. This feedback view is helpful in understanding the development process of crime problems at places. It shows that individuals involved in crime events tend to change their states according to environment feedback. In technical terms of artificial intelligence, this corresponds to learning and adaptation behavior. It also suggests that the simulation of crime events and crime patterns needs to take into consideration the complex feedback interaction and learning behavior.

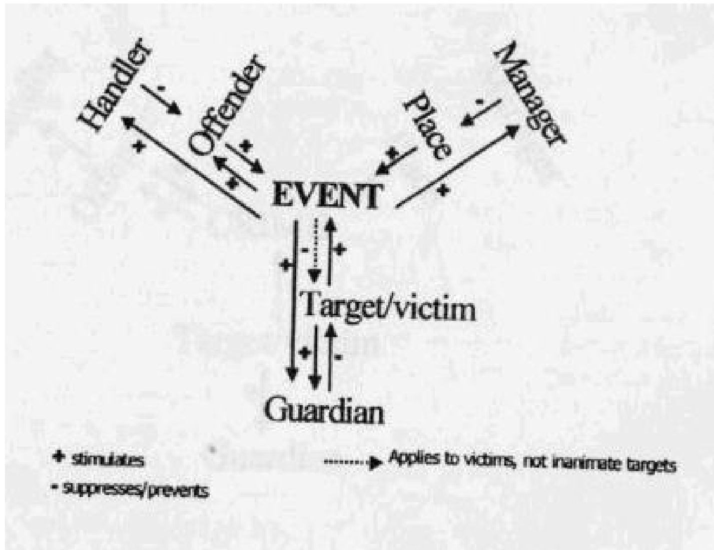


Figure 2.2 Feedback assumption by Eck (2003, p90)

The above review of crime theories depicts a crime event process for crime simulation. Potential offenders, potential targets, and guardians are all involved in their daily routine activities (for example, moving between home, work, shopping and entertainment places). When potential offenders and potential targets meet at the same place and at the same time, there is a possibility for a crime event. However, whether the crime event occurs or not depends on the effectiveness of guardians, handlers, and managers, as well as the status of offenders (e.g., motivation) and targets (e.g., desirability) at that place and that time. The occurrence of a crime event will generate some feedback to offenders, targets, guardians, handlers, and place managers, whose states will change accordingly. The future chance of a crime at the same place will no longer be the same as the previous.

Such a theoretical interpretation of the crime event process provides an opportunity to create a framework for spatial crime pattern simulation. The above review of crime theories indicates that the theoretical background is now ready for such a spatial crime simulation model. However, the spatial crime pattern simulation brings several challenges to spatial modeling and GIS, which include: spatial dynamic process modeling (routine activities over space), cognitive behavior modeling (offender, target and controller spatial learning and spatial reasoning), and integrating such models with GIS. The geographic computational laboratory developed in this research also aims to contribute in these areas. In the following section, methodologies in modeling space-time processes in geography are reviewed.

2.2 GIS-based simulation with CA and agent-based modeling

A top-down approach in geographic modeling often assumes that individuals whose behaviors are to be modeled have complete information, a global view and perfect rationality in their spatial decision-making, or as if there is a central control in directing every individual's decision-making. This approach neglectes an important dimension of spatial behavior: location decision-making is a learned behavior based on positive feedback. Recent developments in geographic modeling methodologies based on cellular automata and agent-based modeling have allowed us to avoid such unrealistic assumptions. Cellular automata and agent-based modeling take a bottom-up approach to model spatial patterns. The assumptions in these two methodologies are more realistic: location decision-making is local, and based on limited knowledge and bounded rationality. The resulted spatial pattern is a self-organized phenomenon without a central control.

This is especially true for crime patterns. As reviewed in the previous section, routine activity theory supports the idea that crime patterns are based on local interaction of offenders, targets, and controllers. Eck's feedback hypothesis suggests that criminal behavior over space is a learned behavior based on positive feedback. This crime simulation modeling aims to show that macro crime patterns result from local interaction and feedback effects among offenders, targets, controllers and the environment. In short, crime patterns are the product of complex systems.

GIS-based simulation is suitable for studying space-time processes that involve a high-level of complexity, such as the spatial interaction between offenders, targets and controllers. However, GIS has been configured as a spatial database management system. The space model (which is absolute space) and data model (which does not include temporal dimension) associated with current GIS technology does not allow us to efficiently model space-time processes within current GIS framework. Instead, the modeling of space-time processes in GIS mainly relies on the integration with cellular automata (CA) and agent-based modeling. CA models have been used in ecological modeling (Maxwell and Costanza 1997; Balzter, et al. 1998; Rajar, et al. 1997), urban growth simulation (Batty and Xie 1999; Clarke and Hoppen 1997; Clarke and Gaydos 1998), forest fire spread simulation (Clarke, et al. 1995), and transportation simulation (Wahle, et al. 2001; Blue and Adler 2001), etc. Agent-based modeling has been used in recreation behavior simulation (Gimblett, et. al. 2002) and building geographic computational laboratories to study the globalization process in spatial network (Dibble, 2001). This research uses agent-based modeling and CA as the tool for modeling the complex space-time processes of routine activities and crime events. In the following section, the basic concepts of CA and agent-based modeling are reviewed.

2.2.1 Cellular Automata

Introduction

The idea of Cellular Automata (CA) dates back to the 1950s when Von Neumann worked on self-producible computers. The basic argument by Von Neumann was that a set of rules or instructions could be found to enable the software of computers to reproduce the structures of themselves. This led to the notion of Cellular Automata, which basically means that local interactions between neighborhood cells based on simple rules can give rise to global structures.

A famous example of CA is the game of "Life" that illustrates key elements of a CA. A "Life" game can be played on a two-dimensional cellular space. There are a set of simple rules that governing the life and death of each cell on this cellular space. For example, a dead cell becomes alive if there are exactly three living cells adjacent to it; a living cell stays alive if there are two or three living cells adjacent to it; a living cell dies from isolation if there are less than two living cells adjacent to it, or it dies from over-crowding if there are more than three living cells adjacent to it. The "life" example demonstrates several key elements of CA, such as states, neighborhood space, cellular space, and transition rules.

Definition of CA in spatial modeling

A CA spatial model has four basic elements, which form a tuple (X, S, N, f). X is cellular space, a set of cells that constitute the space of a study area. X can be one-dimension, two dimension or three-dimension. In spatial modeling, it is typically two-dimension. S is a nonempty finite set of states for each cell in the cellular space. Each cell at a given time can be in only one state. N defines a neighborhood template for a given cell. There are two types of neighborhoods for CA models: Neumann neighborhood (with 4 neighboring cells), and Moore neighborhood (with 8 neighboring cells). Typically a cell has eight neighboring cells in a CA model. The transition function f consists of a set of rules that govern the transition of cell state according to the states of each cell in the neighborhood template at a previous iteration. Wu (1999) defines a two-dimension CA model as:

$$S_{rc}^{t+1} = f(S_{rc}^{t}, N_{rc}^{t}) \ldots\ldots (2.3)$$

where S_{rc}^{t+1} is the state of the cell at time t+1 and at cell (r, c), S_{rc}^{t} is the state of the cell (r, c) at time t, N_{rc}^{t} is the neighborhood space of cell (r, c), and f ( ) is the transition function. Equation (2.3) defines a CA model as the mapping of cell states from neighborhood template at iteration t to the current cell at iteration t+1. Therefore, space and time are all included in the model.

Neighborhood effects in CA models

The usefulness of CA for spatial modeling lies in its capability of including neighborhood effects. CA models are used to simulate the complex space-time process of the emergence of global patterns resulting from local interactions. The neighborhood effect represents the situation for a site. It also reflects a fundamental law in geography: things that are near influence each other. Traditional urban and regional models that are based on global variables are not able to take local situation into consideration. In urban growth simulation, CA models regard urban land use patterns as an emergent property of distributed and local decision-making.

According to Couclelis (1997), GIS has been difficult to apply in urban and regional models. The current space model implemented in GIS is an absolute space model. Absolute space model means that GIS only records the absolute position of a site, but does not explicitly record the situation of this site (for example, the distance between other sites and the current site). Many urban and regional models are based on a space model known as relative space (such as the gravity model of spatial interaction). The relative space model describes the location of a site through its position relative to other sites (i.e., its situation). For example, how far away one site is to other sites. Couclelis (1997) suggests that it is the discrepancy between the underlying space model of GIS and urban/regional models that makes the integration of the two difficult.

The introduction of CA models into GIS and spatial modeling has changed this limitation. The neighborhood effect in CA allows spatial models to address the immediate neighborhood situation in urban modeling. Couclelis (1997) defines "proximity space" as a new space model brought by CA that bridges absolute space and relative space. The introduction of the proximity space model to GIS by CA expands the modeling capabilities of GIS.

In this research, following the RA/CA model, CA is used to model the neighborhood effects of crime events (the occurrence of crime events at one place affects its neighborhood places).

2.2.2 Agent-based modeling

Introduction

Cellular automata models have been used to model neighborhood interaction and to show how local interaction gives rise to global spatial patterns in urban regional studies. The basic unit in CA modeling is a land cell. Land cells are typically not mobile. Therefore, CA mainly models interactions among land cells, but is not suitable to model the interaction among people and between people and environment.

For the past two decades, agent-based modeling (ABM) has been used by social scientists and geographers to model human-environment interactions. ABM supports the modeling of mobile agents in a spatial environment, and allows modelers to specify interaction rules for those agents. Mobile agents can be used to represent offenders, targets, and controllers in crime events. Also, agent spatial learning and adaptation can be integrated into an agent framework. So ABM, coupled with CA, provides the ideal methodology to build the computational laboratory for crime simulations in this study.

SWARM provides a set of class libraries for building agent-based models. A basic agent-based model built in SWARM consists of a population of randomly roaming agents in a spatial environment that interact with each other following a set of rules. Modelers then can observe the emergence of certain patterns resulting from agent interactions.

Definition of ABM in spatially integrated social science

Agent-based modeling belongs to a branch of artificial intelligence technology – distributed artificial intelligence (a collection of methodologies for problem solving that focus on distributed computation). Agent-based modeling is also known as individual-based modeling and an artificial society (Epstein and Axtell, 1996), and agents sometimes are referred to as mobile automata. For the past two decades, social scientists have applied this tool for social simulation. For example, Epstein and Axtell (1996) have built agent-based models to simulate skewed wealth distribution pattern and epidemics in an artificial society.

The following is a definition of agent-based modeling given by Epstein and Axtell (1996) that fits the model developed in this research. Epstein and Axtell (1996, p4) argue that an agent-based model or artificial society includes three basic ingredients – agents, environment, and rules.

Agents are entities of the real world whose behaviors are to be modeled. An agent can be a person, a vehicle, or a household, etc. Physically, agents can be understood as software robots residing in computer memory space that is allocated to an agent-based model. Each agent has internal states, and is able to act according to a set of rules. The action of agent may lead to the change of its states as well as rules that govern its decision-making. In other words, the agent is able to evolve in their interaction with the external environment.

The environment provides space for agents to act in. The environment also should include all other relevant information for agent decision-making, for example, the states of other agents that are within a certain distance of the current agent. The environment could be a geographic landscape (either ideal landscape created by the researcher, or natural landscape observed in the real world).

Rules define the interactions between agents and environment. Rules can be manipulated to test their effects on the global patterns.

According to Epstein (1999), agent-based modeling is a "third" research approach for social scientists, distinguished from the traditional "deductive" and "inductive" approaches. Epstein (1999) named it as a "generative" approach. An inductive approach starts with data and tries to find patterns in the data. A deductive approach starts with basic axiomatic assumptions and tries to find the logical consequences of such assumptions. A generative approach combines the characteristics of both a deductive approach and an inductive approach, but is different from both. The generative approach, like a deductive approach, starts with axiomatic assumptions; and like an inductive approach, generates data for finding patterns. But unlike the deductive approach, it does not directly end up with logical consequences. Unlike the inductive approach, the data in analysis are artificial and are not real world observations. The generative approach uses computer simulated experimentation to study the behavior of a complex adaptive system. Agent-based modeling provides a natural tool to build a laboratory for such experiments.

Characteristics of agents

The term "agent" requires more considerations here, because it is the most basic element in an agent-based model and the characteristics of the agent represents the main advantages of agent-based modeling. According to Weiss (1999), there are three characteristics of agents in a multi-agent system:

Autonomy - each agent can be seen as a software robot. Once an agent is created and released in an agent-based model, it controls its own states and behavior in its environment in order to achieve its designed objectives (such as finding the optimal path on a street network) without intervention from the modeler. Hence, its behavior is unpredictable from the modeler's perspective.

Intelligence - agents can optimize some performance measure (such as happiness) while executing tasks. This is achieved by designing specific rules for agents. For example, in recreation behavior simulation, rules can be given to hiker agents so that those agents avoid areas with high density of other hiker agents. Cognitive behavior, such as learning and reasoning can also be given to agents. For example, in crime simulation, reinforcement learning algorithm can be used to equip agents so that they can learn to perform routine activities (e.g., going to school, entertainment, shopping) within a spatial environment.

Interacting - "interacting" means that actions of one agent are affected by actions of other agents. This is the major source of complexity in an agent-based model. That agents can be designed to interact with each other is a major advantage of agent-based modeling for complex adaptive system because such interactions are difficult to capture by regular approaches.

Reactive agent and cognitive agent

According to Jiang & Gimblett (2002), agents can be classified as reactive agents and cognitive (or deliberative) agents in urban environmental modeling. Reactive agents can only respond to environment states (including states of other agents). They don't remember past experience, and they do not have a model of the environment. Hence, reactive agents could not plan their future activity. Cognitive agents have the capability of learning and adaptation. They have a model of the environment, they have a memory to remember past experience and they can use their experience to improve their future behavior. In the RA/CA crime simulation model, offenders and targets are reactive agents without spatial learning and adaptation capabilities. In the RA/CA/ABM computational laboratory, offender agents and target agents are designed as cognitive agents with spatial behavior adaptation capabilities.

2.2.3 Integration of CA/ABM with Geographic Information Systems

In terms of functionality, CA/ABM complement GIS. GIS has been conventionally configured as a spatial database management system, without effective space-time dynamic processes modeling capability. Cellular automata is able to model the complex space-time processes that involves local interactions and neighborhood effects between land cells. Agent-based spatial modeling is able to model the interaction among a population of agents as well as their interaction with the environment. Because of such complementarities in functionality, the integration of CA/ABM with GIS is desirable.

Such integration is also possible because CA and ABM are cell-based computation models and resemble cell-based GIS in terms of data structure. There are three ways to integrate CA/ABM with GIS: implement CA/ABM model within a GIS, implement GIS functionality within CA/ABM models, and loosely couple GIS with CA/ABM models.

For CA models, the advantage of integrating with GIS is to make rich spatial representations accessible to CA models. Takeyama (1997, p73) developed GeoAlgebra for CA so that the transition rules of CA models can be implemented within GIS. Practically speaking, most CA spatial models are developed outside of GIS and then loosely coupled with GIS (e.g., Batty, Xie & Sun, 1999).

Similarly, for agent-based models, the advantage of integrating with GIS includes making use of spatial data layers. Another advantage is to enable agents with spatial analytic capabilities. For example, an agent may need to calculate distance or cost from origin to destination. However, due to the technological limitations of commercial GIS (which will be discussed later), no agent-based model has been directly implemented in GIS. Instead, agent-based spatial simulation models are often built outside of GIS, and then coupled with GIS. GIS data can be imported to these agent-based models, and the output of simulation can be visualized and analyzed in GIS (for examples, see Gimblett, et. al. 2002; Itami, 2002).

2.3 The RA/CA crime simulation model

This research is partly based on the RA/CA crime simulation model developed by Liu et. al. (2004) and Liang, et. al. (2001). This section will present a brief review of the RA/CA model, including its contributions and limitations in crime simulation.

The RA/CA crime simulation model is the first attempt to build a computational model for crime patterns. The model includes two versions: the RA/CA model for commercial robbery by Liang, et. al. (2001), and the RA/CA model for street robbery by Liu, et. al. (2004). The basic structures of the two versions are the same except that the crime types modeled are different.

In RA/CA model, the space-time dynamic processes are modeled by Cellular Automata coupled with a Monte Carlo simulation. A Monte Carlo simulation is used to model the activities of offenders and targets (targets are only mobile in the street robbery case). Offenders are assigned home locations, and their interactions with crime places are controlled by a mean information field. The mean information field stores the probabilities of visiting crime places for every offender. The mean information field is constructed so that locations near the offender's home have higher chances of being visited by the offender. For the street robbery case, target activities are also controlled by mean information field except that targets are not assigned home locations, and the chances for target to visit each location are equal on the whole street network. As will be discussed later, the Monte Carlo simulation represents several limitations for the RA/CA model, which will be improved in the current research.

In RA/CA model, the Monte Carlo simulation determines the distribution of offenders and targets in space. The convergence of offenders and targets at suitable places create crime events. The crime event likelihood is evaluated by Eck's formula (Eck, 1995). The factors that affect the crime event likelihood include target desirability, guardian capability, offender desirability, place management effectiveness, and place accessibility.

In order to model the spatial effect of crime events at neighborhood places, a Cellular Automata model is built to model spatial diffusion process of tension variable. Tension is the state variable of the CA model. Tension (Liang, Liu & Eck, 2001) is defined as psychological reaction of places to crime events, which is similar to anxiety or fear of crime. The concept of tension is an important contribution of RA/CA model and will still be used in the RA/CA/ABM computational laboratory.

In RA/CA model, offender motivation can evolve according to experience of success or failure on offending. Through the tension variable, target properties and place properties can be adapted after being attacked. However, their spatial activities are not adaptable to crime events. No matter how many successes or failure an offender has at a place, the offender does not change the probability of attacking that place. Similarly, no matter how many times a target is attacked in a certain place, the target will not change her activity pattern. According to the feedback hypothesis, offenders would be reinforced by past successful experiences at one place, and increase their presence at that place; targets will be discouraged to go to places where they are repeatedly attacked. Therefore, the RA/CA is not able to simulate the dynamic space-time distribution of offenders and targets, even though it can be applied to study the evolution processes of offenders and targets.

These limitations of RA/CA model are caused by Monte Carlo simulation. With Monte Carlo simulation, the movement of offenders and targets are determined by random numbers and probability fields. Therefore, people can "jump" from one place to another because the next place a person will be is determined by a random number compared with a mean information field that contains all places. Not only this is not realistic, but also it leaves spatial learning and adaptation of spatial routine activities difficult to model.

The components of the RA/CA model are summarized in Figure 2.3. At each iteration of the simulation, the model starts with the Monte Carlo simulation of offenders and targets movement. Each offender and target is assigned a location. Then the statuses of offenders, targets and places are taken into consideration to evaluate crime event likelihood. If a crime event occurs, then tension at the place will be increased and diffused to other places through a CA model. The broadcasted tension will affect the properties of offenders, targets and places. At each iteration, target, offender and place properties will be adapted according to what happened at last iteration.

Built on the theories, methodologies, and previous work reviewed in this chapter, the conceptual framework of the RA/CA/ABM computational laboratory is presented in the next chapter.

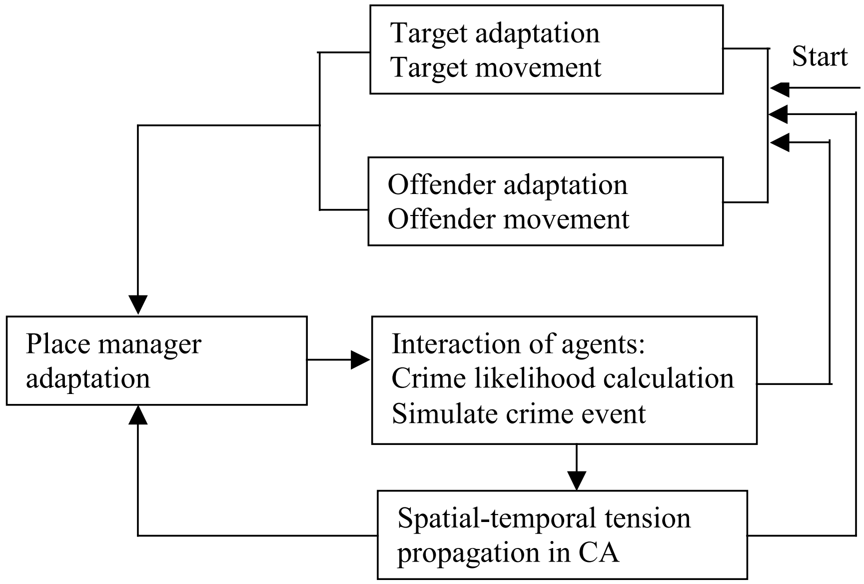


Figure 2.3 Major components and interconnections of RA/CA model (modified from Liu, et. al. 2004)

Chapter 3 The RA/CA/ABM Computational Laboratory

3.1 Introduction and basic assumptions

This chapter presents the conceptual design of the RA/CA/ABM computational laboratory in an agent-based framework. The computational laboratory provides a crime event and crime pattern simulation model that can be customized by environmental criminologists to make their own experiments. The simulation model is built according to the crime event model of routine activity theory, incorporating offender agent, target agent, police agent, place agent and their interaction and feedback rules. The offender agent, target agent and police agent are derived from a base class of a spatial autonomous agent. The spatial autonomous agent is built using a reinforcement learning algorithm. Crime event rules are created for offender agents, target agents, police agents and place agents.

This chapter first presents the design of a spatial autonomous agent for spatial activity simulation, and then presents the design of the offender agent, target agent, police agent, and place agent. Adaptation rules of offender agents, target agents and place agents are described. Next, crime event rules are presented. Finally, the framework of the spatial crime event and crime pattern simulation model is presented as a summary to the chapter.

In order to reduce the complexity of the simulation modeling to a manageable level, the following assumptions based on crime event theories need to be made before formally presenting the model.

(1) Considerations on crime types in the simulation

There are many different types of crime. The broad categories include property crime and violence crime. Different crimes require different explanations. Because our goal of simulation modeling is to study how micro interactions between offenders, targets and crime places give rise to macro crime patterns, the model will become unnecessarily complex if we include several crime types simultaneously in the model. Instead, this simulation model includes only one type of crime – street robbery. The advantages of choosing street robbery include: (1) street robbery is a type of direct-contact predatory violation (which is defined as "at least one offender take or damage the property of at least one other person by force or threat of force" (Clark, et. al. 1993)), which is the typical crime type explained by the original version of routine activity theory; (2) in street robbery, targets and offenders are all mobile, unlike burglary or commercial robbery.

(2) Scale of the simulation

This simulation modeling is interested in crime patterns at micro crime places, which is at a sub-neighborhood scale. Therefore, for theoretical simulation purposes, the simulation does not need an area that is larger than several blocks in a city. Even though a larger area of study could also be used, it will consume too much computational resource. (Agents are modeled as cognitive agents in this simulation study. Each agent consumes a lot of computer memory.)

(3) Spatial-temporal resolution of the simulation

Agents occupy space and time to run routine activity schedules. Therefore, space and time are all converted into discrete representations. This is done by converting vector representation of the study area into grid data in GIS. Crime places are individual cells. The cell-based street network also serves as the routine activity space for agents. Because the simulation is interested in crime patterns at crime place level, cell size is set at the address level – 20 feet (about 6 meters). The length of iterations in the simulation is set to 15 seconds, which is about the time that a street robbery event requires.

(4) Routine activities and offender search assumptions

Routine activities have a hierarchical structure (Golledge & Stimson, 1997). The routine activities of a typical person may include: staying at home, going to work, shopping, and entertainment. These are the main activities. However, there are also some other activities in one's daily life that accompany these main activities. For example, some people may commit crime when involved in entertainment activities. The activity patterns of these activities are structured by main activities.

Trips associated with certain routine activity can have multi-purposes (Golledge & Stimson, 1997). For example, offender agents go to a shopping store both for a shopping purpose and for a criminal purpose. This research assumes that offenders commit crime while in other legal purposes (such as shopping). This assumption is especially true for opportunistic offenders who often conduct criminal activity during legal routine activities (Brantingham & Brantingham, 1993).

(5) Rational choice assumptions

Following rational choice theory (Clark & Felson, 1993) of crime, this simulation modeling assumes that an agent has basic rationality. For example, target agents avoid cost and pain, and agents will follow optimized paths (with minimum cost) to reach a destination. The cost for a target agent includes cost of distance and cost of crime (which is defined as a negative reward received from environment). For an offender agent, the cost is by distance friction. Different from a target agent, an offender agent receives positive rewards from environment, and for a rational offender, the offender tries to minimize distance cost and maximize positive rewards from crime.

During routine activities, potential targets and potential offenders all try to minimize costs while maximize rewards. This defines the most basic spatial intelligence for offender and target agents.

3.2 Spatial autonomous agents for routine activity simulation

Routine activity theory tends to treat people as objects with space-time positions (Clark & Felson, 1993). Crime patterns are structured by routine activity patterns of offenders and targets. Therefore, routine activity simulation is critical to crime pattern simulation.

The crime simulation models use agents to represent offenders, targets, and police officials. The common routine activity functions required by offender agents, target agents and police agents are designed within a basic type of spatial autonomous agent. This spatial autonomous agent is then used to derive offender agents, target agents, and police agents.

Geographers use spatial autonomous agents to build spatial simulation models (e.g., Gimblett, 2002). However, there have not been enough studies in spatial autonomous agent to give it a formal definition. According to Weiss's (1999) definition of an autonomous agent (which is a computer program being able to achieve designed objective without external control), in this research, we may give spatial autonomous agent an interim definition as "a computer program that is able to achieve a spatial goal (such as finding way from origin to destination on a street network) without external control".

3.2.1 Defining agent routine activities

Daily activities as routines are defined as "a recurring set of episodes in a given unit of time" by Golledge & Stimson (1997, p290). For example, a person's daily activity routine may include:

going to work, shopping, and going back to home. In this case, there are three episodes of activities in a one-day period: from home location to work location, from work location to shopping location, and from shopping location back to home. Therefore, the simplest routine is a one-episode activity (the movement from an origin to a destination (see Figure 3.1)). Any other complex activity routine in a given period of time can be seen as the composition of several episodes (see Figure 3.2).

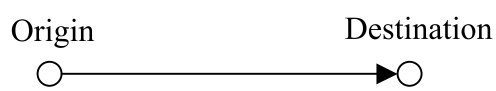


Figure 3.1 The simplest routine activity (containing only one episode, arrow indicates agent movement direction)

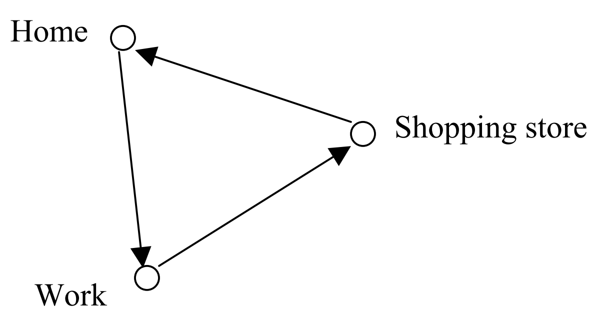


Figure 3.2 Routine activities containing several episodes (arrows indicate agent movement direction)

As models of people, a spatial autonomous agent is designed with the capability of executing routine activities in an urban space. In order to execute routine activities, agents first need a routine activity schedule, and then need the navigation and way-finding capability. Furthermore, for crime event and crime pattern simulation purposes, agents need to be able to learn and adapt from past crime experiences and change their future routine activities. Agent routine activity schedules and wayfinding capability are presented in the following sections.

3.2.2 Agent routine activity scheduling

Each agent needs a schedule of routine activities to act on the street network. The routine activity schedule completely defines an agent's activity in a given period of time. Routine activity schedules of individuals in the real society have been studied by transportation planners (Golledge & Stimson, 1997). In these studies, duration and sequence of activities are used to describe routine activity schedules. Duration means that people allocate a certain amount of time for each activity. Sequence is the order in which activities are carried out.

This research develops an agent routine activity schedule that specifies a sequence of activities for each agent. The routine activity schedule takes the form of a matrix, with each row of the matrix stores the transition probabilities of the next activity. The transition of one activity to another activity is not always fixed (Golledge & Stimson, 1997). For example, the sequence of going to shopping and going to a bar is not always fixed. When an agent is created, it is initialized with an activity. When the model runs, the agent looks up the matrix to decide its subsequent activity.

Activity1\Activity2	a1	a2	a3
a1 (home)	p11	p12	p13
a2 (work)	p21	p22	p23
a3 (shopping)	p31	p32	p33

Figure 3.3 Agent activity schedule for the routine presented in Figure 3.2

An example of such a schedule for the daily routine of Figure 3.2 is illustrated in Figure 3.3. In Figure 3.3, a1 represents "going home", and (p11, p12, p13) represents transitional probabilities for activity a1, where p11 + p12 + p13 = 1.0. The probability for activity a1 to be the next activity of a1 is p11, and the probability for activity a2 to be the next activity of a1 is p12, and the probability for a3 to be the next activity of a1 is p13. The next row of the matrix is a2, representing “going to work”, where (p21, p22, p23) represents transitional probabilities following activity a2. The third row of the matrix is a3, representing “going to shopping”, and the same rules apply.

When an agent is created, the agent is initialized with an activity, for example, a1 (going home). Figure 3.4 presents a configuration of the activity schedule in Figure 3.3. Here is how this schedule defines the movement of agents in the circle of home, work, and shopping location. After the agent finishes the activity of “going home”, the agent decides next activity using the transition probabilities (p11 = 0, p12 = 1, p13 = 0). Because p11 and p13 both equal 0, and p12 equals 1, next activity is chosen as “going to work”; after the agent reaches work location, then the agent decides next activity based on transition probabilities (p21 = 0, p22 = 0, p23 = 1). Because p21 and p22 are both 0, and p23 is 1, the next activity is chosen as “going to shopping store”. After the agent finishes shopping, then the agent chooses next activity as going back home, which is easy to determine from the transition probabilities (p31 = 1, p32 = 0, p33 = 0).

Activity1\Activity2	a1	a2	A3
a1 (home)	0	1	0
a2 (work)	0	0	1
a3 (shopping)	1	0	0

Figure 3.4 A configuration of routine activity schedule in Figure 3.3

It is worth noting that the routine activity schedule developed for agents can be extended to include more activities than three.

3.2.3 An artificial intelligence model of agent wayfinding

Finally, agents need the capability to execute a routine activity schedule. The capabilities to execute a routine activity schedule include two parts: first, agents should be able to find way to destination. Second, agents need to be able to change to new routes according to an experience of crime. According to Golledge & Stimson (1997), the individual's perception of environment changes via complex learning processes. This may lead to the changing of routine activity patterns (including switching route and destination). Crime pattern theory argues that individuals develop awareness spaces during routine activities (Brantingham & Brantingham, 1993). Golledge & Stimson (1997) also argue that there is a learning process accompanying routine activities – individuals modify their perception of the environment by moving within it and by communicating about it with their peers.

This section applies a reinforcement learning algorithm to model agent wayfinding and learning process. This wayfinding and learning model enables an agent with spatial knowledge acquiring, spatial reasoning, and activities adaptation capabilities when the agent moves within a spatial environment and communicates with other agents.

As stated earlier, the simplest routine activity is a single-episode routine activity. In this section, we will describe how agents learn to find their way from origin to destination in executing a single-episode routine activity. When agents are created, they are initialized with a few properties (such as guardian capability and desirability), and an initial position on the street network (such as home location or just a random position). They have no spatial knowledge about how to reach the destinations. With a reinforcement learning algorithm, agents are designed to be able to learn distance cost and directions to the destination through a large number of interactions with the environment. The learned knowledge is then used to guide agents to perform routine activities. Agent learning accompanies every step of movement, so that future change of the environment (such as the occurrence of crime events) can be reflected in agent spatial knowledge and in turn changes their spatial activity pattern.

A. Agent spatial knowledge representation

Agent spatial knowledge is stored in a cognitive map. A cognitive map can be understood as a model that an individual constructs for the environment (Golledge & Stimson, 1997). A cognitive map stores an individual's spatial knowledge, such as distances, directions and relative positions of spatial objects. In this study, cognitive maps serve as working memories for agents, directing agents' spatial movement. On the other hand, a cognitive map is the result of an agent's learning during spatial movement.

The cognitive map is modeled as a least cost surface associated with a destination. Each cell on the grid stores the accumulated least cost from that cell to the destination cell. This least cost value is learned by an agent when the agent moves within the street network and communicates with other agents, and the least cost values are used in guiding agent's movement. An example of agent cognitive maps is presented in Figure 3.5a and Figure 3.5b.

Because each cognitive map provides information for one activity, if an agent routine has several episodes, then the agent needs several cognitive maps, one for each activity. Thus, agent spatial knowledge is represented in several cognitive maps. Such agent spatial knowledge structure is illustrated in Figure 3.6.

28.46	25.46	24.21	22.97	21.73	20.49	19.24	18.00	19.24	20.49
27.21	24.21	21.21	19.97	18.73	17.49	16.24	15.00	16.24	17.49
25.97	22.97	19.97	16.97	15.73	14.49	13.24	12.00	13.24	14.49
24.73	21.73	18.73	15.73	12.73	11.49	10.24	9.00	10.24	11.49
23.49	20.49	17.49	14.49	11.49	8.49	7.24	6.00	7.24	8.49
22.24	19.24	16.24	13.24	10.24	7.24	4.24	3.00	4.24	7.24
21.00	18.00	15.00	12.00	9.00	6.00	3.00	0.00	3.00	6.00
22.24	19.24	16.24	13.24	10.24	7.24	4.24	3.00	4.24	7.24
23.49	20.49	17.49	14.49	11.49	8.49	7.24	6.00	7.24	8.49
24.73	21.73	18.73	15.73	12.73	11.49	10.24	9.00	10.24	11.49

Destination

Figure 3.5a An example of agent cognitive map (each cell stores accumulated least cost value to the destination cell)

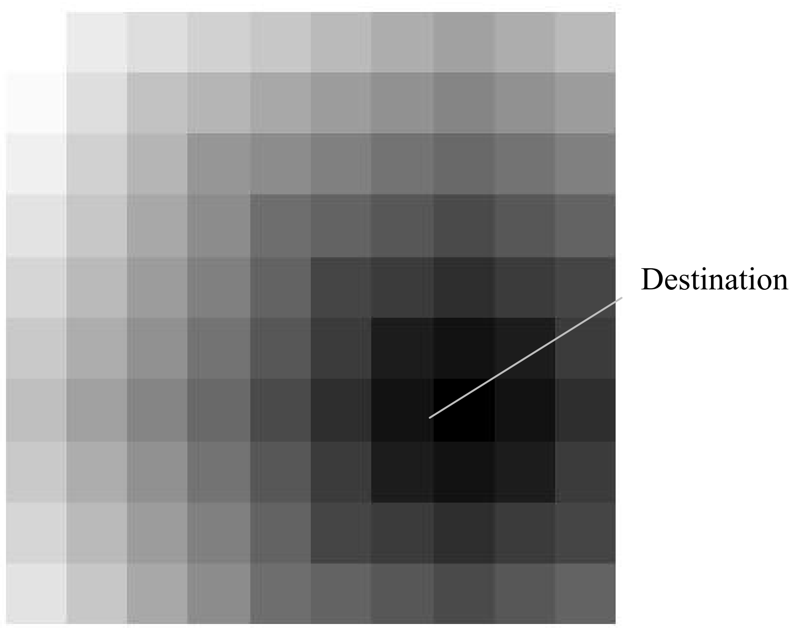


Figure 3.5b A rendered view of the grid in Figure 3.5a

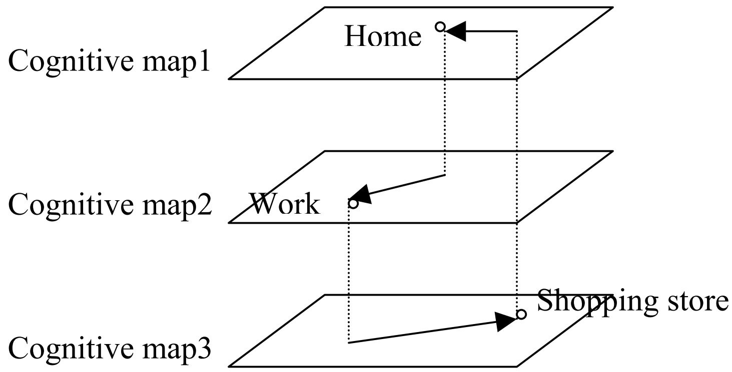


Figure 3.6 Agent spatial knowledge structure (for executing routine activities illustrated in Figure 3.2, each map guides the agent to one destination)

B. Agent spatial reasoning

With the least cost surface as agent cognitive map, it is easy to derive movement directions used by an agent to reach the destination. Suppose that the agent is at cell (i, j) (i and j are row and column number on the grid). Then the agent has eight choices of next position, corresponding to the eight neighborhoods NE (i+1, j+1), E (i, j+1), SE (i-1, j+1), S (i-1, j), SW (i-1, j-1), W (i, j-1), NW (i+1, j-1), N (i+1, j).

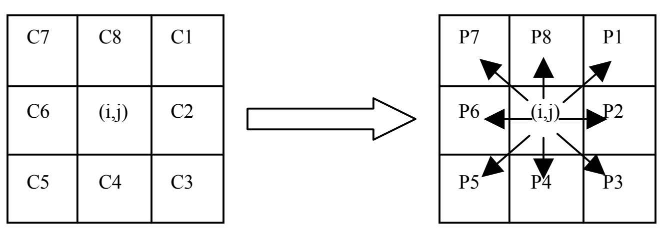


Figure 3.7 Agent spatial reasoning – mapping the accumulated cost values to action probabilities

Now the agent must make a decision on which cell to step into. The agent makes use of the information on the cognitive map. Because each cell of the cognitive map stores accumulated least costs to the destination, the agent is able to retrieve the least cost values for the eight neighborhoods (C1, C2, …, C8 in Figure 3.7). Now, we assign some probabilities for the eight neighborhood cells as agent action probabilities (P1, P2, …, P8 in Figure 3.7) according to their accumulated costs to destination. According to the rational choice assumption that has been stated at the beginning of this chapter, the agent would prefer the neighborhood with the smallest cost value. For example, we assign 100% probability to the cell with smallest cost value, and 0% to all other cells. Then, the agent will choose the cell with the smallest cost value to step into (we call this "greedy action"). Thus, the agent will follow the least cost path to the destination. However, a strict least cost path may not be always reasonable. Human beings are not always following the optimal paths in reaching their spatial goals. Thus, instead of assigning 100% probability for the cell with smallest distance value, we gave slightly less probability to the greedy action, e.g., 90%. The rest 10% probability is assigned to other cells.

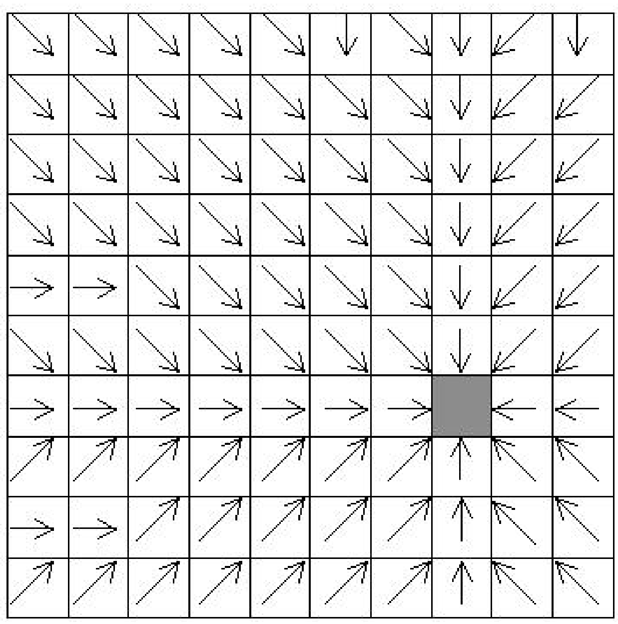


Figure 3.8 Agent movement directions according to cognitive map in Figure 3.5a

Based on such spatial reasoning framework and agent cognitive map, the agent is able to find its way to the destination, no matter where the agent is. Each step of the reasoning moves the agent one step forward. For the cognitive map presented in Figure 3.5a, agent movement directions (if the agent takes greedy action each step) are illustrated in Figure 3.8.

C. Agent spatial learning

When an agent is created, it has no spatial knowledge, and the agent cognitive map is initialized by a uniform value for all cells. The movement of the agent would seem random at first. There are potentially two approaches for an agent to get the cognitive map (the least cost surface). One approach is use graph theory-based algorithm to calculate the least cost values for each cell, like the algorithm used by ESRI spatial analyst extension. The other is using a machine learning algorithm. This research uses the machine learning approach because of the following considerations. First, agents should not have perfect knowledge within the whole simulation area. According to crime pattern theory, offenders develop awareness space around major activity nodes and paths. Other places that are far away from routine activity nodes and paths are not well known. If we use graph theory-based algorithm, then the least cost surface is perfectly developed over the whole simulation area, which can not be realistically obtained by individual agents. Second, a machine learning algorithm can be integrated with agent movement, so that the learning is smooth and associated with agent experience.

A spatial learning algorithm is designed for the agent to learn the cognitive map through interaction with the environment. When an agent moves in an environment, it keeps estimating the least cost values for the cells that it visited and updating the cognitive map with those values. Thus, an agent would improve its performance through a large number of trials. The movement would gradually form a fixed pattern, which is close to a least cost path. Also, agents are supposed to adapt routes or destinations when the environment changes (for example, a target agent perceive high crime risk at some place). This requires agents to learn in real time (all the time when they move in the environment).

A reinforcement learning algorithm is used as an agent learning algorithm in this research. In the following, first the general reinforcement learning theory is reviewed, and then how the Q-learning algorithm of reinforcement learning can be used to model agent spatial learning is discussed.

<u>(1) The general reinforcement learning theory</u>

<u>Overview</u>

Reinforcement learning is a group of machine learning methods that enable a computer program to learn to achieve some goal through interacting with environment. For example, reinforcement learning algorithms have been used to design computer programs that can be trained to play chess. Recent years, there have been such programs that can beat human champions in the world. (For a relatively complete introduction to reinforcement learning, readers are referred to Sutton & Barto, 1998).

Reinforcement learning is a new approach in building spatial intelligent agents. Rodrigues (1999) first tried to apply reinforcement learning algorithm to learn spatial concepts and properties. The spatial intelligent agents that Rodrigues developed include GIS interface agent that can adapt to users and help users to better use a GIS, and spatial intelligent agents that can learn to perform spatial tasks in a simulated environment.

At the heart of a reinforcement learning problem is agent learns to achieve goals through interacting with the environment. The learner and decision-maker is an agent; all other things that the agent interacts with belong to environment (Sutton & Barto, 1998). Figure 3.9 is a framework of reinforcement learning.

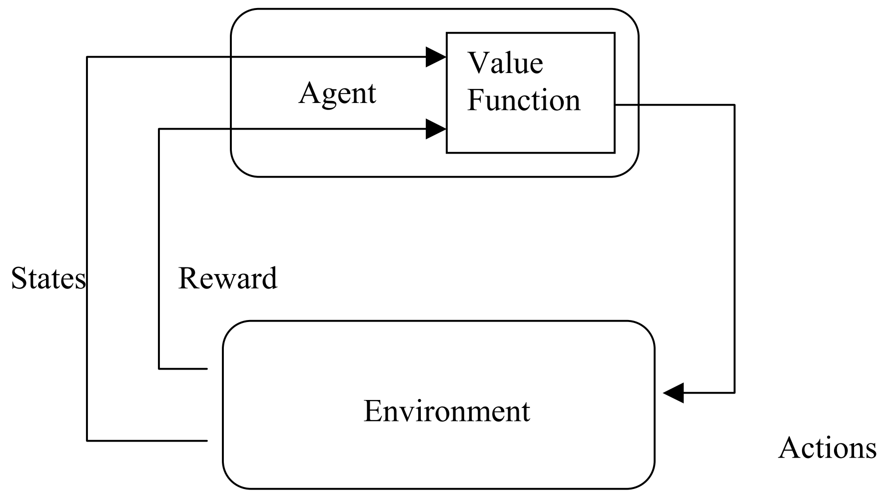


Figure 3.9 A framework of reinforcement learning problem (modified from Sutton & Barto, 1998)

As illustrated in Figure 3.9, a reinforcement learning model includes four elements:

a. A set of environment states;

b. Reward function from environment;

c. A set of agent actions;

d. A value function.

If we classify these elements into two groups – the agent and the environment, then states and reward function belong to the environment, value function and actions belong to the agent. States are some property of the environment that can be perceived by an agent. For example, the temperature of the room is a state. An agent tries to change the environment (for example, change the temperature of a room) through actions (such as change the temperature setting), and the environment feeds back rewards to the agent, thereby tells the agent how “good” that action is. The goal of a learning task is defined by the reward function. The value function is the memory for agent past experience as well as the basis for current agent action decision-making. As will be explained later, the value function is built based on rewards the agent receives from the environment. Typically, a value function stores values for each state or each action, indicating how "good" that state is or how "good" it is to take that action. The key to agent learning is that rewards from the environment are used to improve the value function. After a large number of interactions with the environment, the value function tends to converge to a fixed pattern, and then the learning process is said to be finished. Finding the value function is the major task faced by a reinforcement learning problem, and there are several existing approaches to do this.

Value function

The value function needs to be addressed with more detail because agent cognitive map (the least cost surface) can be treated as a value function of a reinforcement learning problem. The value function stores a utility value for each state or action. A state can be understood as any information from the environment that the agent needs to make decision on which action to take.

The utility value of each state is defined as the accumulation of rewards from a state to the terminal state, as follows (Sutton & Barto, 1998):

$$U(s) = r_1 + r_2 + r_3 + \dots + r_T \quad \dots\dots(3.1)$$

Where U(s) is the utility value for state s, and $r_1, r_2, \dots r_T$ is the reward sequence the agent receives in states $s_1, s_2, \dots s_T$ that follow state s, and r_T is the reward from the terminal state that ends the episode. If the optimized "path" (meaning a sequence of states before terminal states) is not learned by the agent, then the value of U(s) can be very large. There are a number of approaches that can be used to optimize U(s) for all states or actions. These approaches are known as "learning" approaches. (Therefore, a learning approach is actually a computational approach in optimizing the value function)

An example

An example of reinforcement learning application should help the illustration. Reinforcement learning does not have to be applied in spatial simulation. In fact, most applications are in engineering areas, such as robotic control. The example illustrated below is a non-spatial reinforcement learning problem from Sutton & Barto (1998).

Suppose that a reinforcement agent is used in learning to play the chess game "tac-tic-toe". This is a kind of chess with a chess board of 3 rows and 3 columns (see figure 3.10). Typically 2 players are needed for the game, and our reinforcement learning agent is one player (the opponent can be a human being or another computer, for example). Suppose that our reinforcement learning agent plays "O", while the opponent plays "X". If three "O" are in a line (either horizontal, vertical or diagonal), then our agent wins; if three "X" are in a line, then the opponent player wins. If neither player win before the chessboard is filled up, then the game is a draw.

In this reinforcement learning problem, each configuration of the chess board is a state (e.g., figure 3.10 is a state). Each movement of the agent is an action (note that each movement will lead the chess board into a new state). The reward function is decided as follows: if our agent lose (three "X" are in a line), then the agent receive a reward –1; if our agent wins (three "O" are in a line), then the agent receives a reward +1; for all intermediate states before one player wins out, the agent receives a reward 0; if the game is a draw, then the agent receives a reward –1 (meaning that a result of draw is as undesirable as losing).

In order to learn how to act at each state, the agent builds up a value function for all possible configuration of chessboard (states). The value function is in a table. Each entry of the table corresponds to a state. Each state value stored in the table is the accumulated reward following that state (as equation 3.1), which is then averaged over a number of episodes. For example, for a state with three "O" line up (which is a terminal state), the state value is 1, because the agent always wins at that state. For an intermediate state (such as the state in figure 3.10), the state value is the immediate reward (which is 0) plus all rewards the agent receives after that state until the episode end, and then the value is averaged over many episodes to get a precise estimation of the state value (Note: there are several ways to estimate the value besides a simple average. This simple average method is known as "Monte Carlo" approach in solving the value function). Because an intermediate state may lead to win or lose, the averaged state value should be less than 1 and greater than –1. The greater the state value is, the more desirable that state is to move in. Once the value function is optimized through large number of episodes, the value function will guide the agent to act at each step with optimized movement. For example, when the agent is in the state of figure 3.10 and the agent wants to know how to move, then the agent looks up the value function, and move to the next state with highest value. This simple chess-playing example illustrates the concepts of states, actions, reward function, and value function.

X		
	O	
X		

Figure 3.10 The chessboard of "tac-tic-toe" game

<u>(2) Agent cognitive map as a value function</u>

Let's return to the least cost surface that is used as an agent's cognitive map. Each cell in a least cost surface stores the accumulated cost value from that cell to the destination cell. The accumulated cost value stored in each cell in a least cost surface can be formulated as:

$$C(s) = d_1 + d_2 + d_3 + \ldots + d_T \text{ ...... } (3.2)$$

Where C(s) is accumulated cost for cell s, and d_1, … d_T are the cost of each movement (for example, distance friction) that follows position s before reaching destination. If C(s) is minimized (or maximized if the cost value C(s) is represented as a negative value), then C(s) is the least accumulated cost value for cell s.

If we compare equation (3.2) and equation (3.1), it is easy to see that C(s) and U(s) is actually the same thing. In other words, accumulated cost value on a least cost surface is actually a utility value in a value function. We should be able to use reinforcement learning algorithms to find the least cost surface. In order to define the least cost surface as a value function that can be learned by a reinforcement learning agent, we need to map the elements of the reinforcement learning agent to a spatial environment, as follows:

a. Value function: the least cost surface that we need to find;

b. State: each cell on the grid that the agent may stand in;

c. Reward function: distance friction grid combined with reward of crime (the reward is a positive value for offender, and a negative value for target);

d. Action: the movement of agent from one cell to another.

With this formulation, it is clear that the least cost surface can be derived by reinforcement learning. Therefore, we design our spatial autonomous agent as a reinforcement learning agent. So far, our spatial autonomous agent has the value function (the cognitive map) and action rules (the spatial reasoning rules) defined. The only thing left to do in order to make it a reinforcement learning agent is to provide a spatial learning algorithm and a spatial reward function. Next, the Q-learning algorithm as the spatial learning algorithm for our spatial autonomous agent is introduced.

(3) Cognitive map learning through Q-learning

In the last section, the cognitive map was shown to be a value function of a reinforcement learning problem. In this section, a reinforcement learning algorithm, Q-learning, is introduced as agent spatial learning algorithm. The modified Q-learning function is associated with agent movement. Therefore, the agent learns a cognitive map when moving on the street network.

Basic Q-learning function

Q-learning is a learning approach for solving reinforcement learning problem. The 1-step Q-learning algorithm is described by formula (3.3) (Sutton & Barto, 1998).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \ldots\ldots(3.3)$$

Where $Q(s_t, a_t)$ represents action value when agent is in state s_t and take action a_t, r_{t+1} is the reward associated with action a_t that leads the agent to state s_{t+1}, α is learning rate, and γ is the discount factor of using future action value to update current action value.

Modified Q-learning function for spatial learning

Formula (3.3) can be used to enable spatial agents to learn an accumulated cost surface. For an agent moving on a grid, each action (movement) lead the agent to a single state, so the action value actually equals the state value (which is the accumulated cost value for each cell). Also we assume there is no discount for future states, so γ is set to 1. Therefore, formula (3.3) is modified to formula (3.4) to enable an agent to learn accumulated least cost values on a grid. The agent executes formula (3.4) to update its cognitive map at each movement.

$$U(s) \leftarrow U(s) + \alpha[r_{s'} + \max_{s'} U(s') - U(s)] \ldots\ldots(3.4)$$

Where $U(s)$ is utility value/accumulated cost value of cell s, and s' is one of the eight neighborhood cells of cell s. α is the learning rate. The $maxU(s')$ allows an agent to choose the maximum value in the eight neighborhoods of cell s and use it to update the value in cell s. (Note:

Cost values stored in the least cost surface are negative. Therefore, *maxU(s')* corresponds to minimum cost value. For example, "-3.5" is used to represent a cost value of 3.5). The spatial learning takes the reward of moving into cell *s'*, $r_{s'}$, into consideration. The reward function is a spatial one, with each cell storing the reward of moving through cell *s*.

Spatial reward function

The value function of Q-learning stores the accumulated reward over a sequence of actions. Offender and target agent spatial reward functions includes two components: distance friction and cost/reward of crime, which is illustrated in formula (3.5).

$$r_{s'} = d_{s'} + e_{s'} \ldots\ldots(3.5)$$

Where $d_{s'}$ is the distance friction caused by moving into cell *s'*, and $e_{s'}$ is the expected reward of crime at *s'* cell that is stored in a crime template. The reward of crime is taken into consideration in agent Q-learning, so that agent can adapt according to their offending/victimization experience. The crime template will be defined in later sections. Distance frictions are stored in a grid while each cell stores the cost of distance caused by passing that cell.

Distance friction grid

All spatial activities are subject to the restriction of distance cost. This restriction is imposed on an agent by providing distance friction as a negative reward for the agent. An agent is rewarded a certain distance friction for each step it moves. The distance friction values are stored in a grid, with each cell storing the cost of distance caused by passing through that cell. This research assumes that the cost of distance is uniform over the whole simulation area. So, each cell stores the same value. Figure 3.11 is an example of such distance friction grid. Let *dc* denote the values stored in the distance friction grid. Then, for $d_{s'}$ in formula (3.5),

$d_{s'} = dc$ if $maxU(s')$ occurs in a 4-neighborhood cell, and

$d_{s'} = SQRT(2)* dc$ if $maxU(s')$ occurs on a diagonal direction.

The process that an agent uses Q-learning to learn cognitive map is illustrated in Figure 3.12.

-0.5	-0.5	-0.5	-0.5	-0.5
-0.5	-0.5	-0.5	-0.5	-0.5
-0.5	-0.5	-0.5	-0.5	-0.5
-0.5	-0.5	-0.5	-0.5	-0.5

Figure 3.11 An example of agent distance friction grid

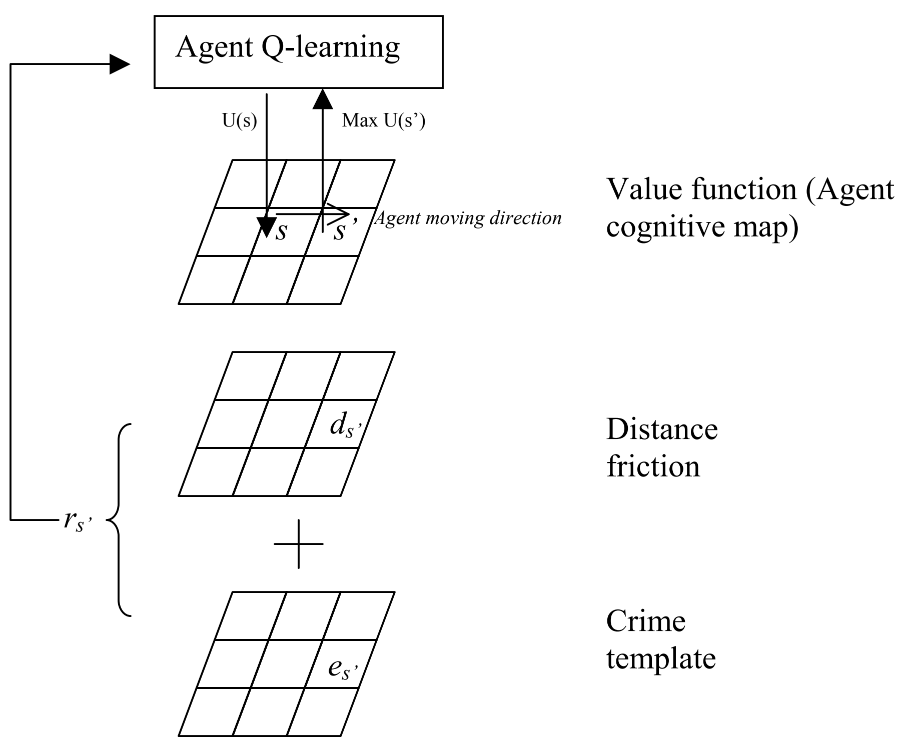


Figure 3.12 Agent estimating accumulated cost values through Q-learning

3.2.4 Summary - the framework of a spatial autonomous agent

Section 3.2 illustrates the design of a spatial autonomous agent that is the base class of offender agent, target agent, and police agent. A spatial autonomous agent has a routine activity schedule, a set of cognitive maps, a set of spatial reasoning rules, and a spatial learning function. A cognitive map is a least cost surface, and a spatial learning function is a Q-learning function.

For a single-episode routine activity, agent wayfinding algorithms are illustrated in figure 3.13.

1. Initialize cognitive map with a uniform float value (e.g., 0.1);
2. Initialize agent with a starting position;
3. Agent updates the least cost value at current position s using equation (3.4);
4. Agent moves a step (according to the spatial reasoning rules);
5. If new position is the destination, go to step 2 (starting a new episode); otherwise go to step 3.

Figure 3.13 Agent wayfinding process in a single-episode routine

3.3 Agents in crime simulations

3.3.1 Offender agent

An offender agent is derived from a spatial autonomous agent presented in the last section. Besides all the capabilities in spatial movement and spatial learning, offender agents are designed with properties that allow them to commit crime during routine activities.

Offender agent properties

According to Eck's (1995) formulation of a crime event likelihood, offender motivation is an important factor for a crime event to occur. Also, for offender agents, learning and adaptation (non-spatial) from past experience, offender agents need to remember their past successful and failed offenses. Also, a learning function is defined for offender agents to develop their motivation following offenses. In summary, offender agents have the following properties:

Property	Data type	Functionality
(row, column)	Point	Record offender agent current position
Cognitive maps	Pointer to grid array	Storing spatial knowledge for routine activities
Activity schedule	Transition matrixes	Scheduling offender routine activities
Distance friction	Pointer to a grid	Cost of distance by moving through each cell
Offending template	Pointer to a grid	Spatial memory of past offending experience
Motivation	Numerical	Propensity in offending
NumSuccess	Numerical	Count of past successful offenses
NumFailure	Numerical	Count of past failed offenses
Ku	Numerical	Steepness of the S-curve learning function

Offender adaptation through offending experience

According to the feedback assumption by Eck (2003) and offender learning suggested by crime pattern theory (Brantingham & Brantingham, 1993), this research assumes that offender agents change their spatial behavior and non-spatial properties following each offense.

(1) Offender spatial activity adaptation

As suggested by crime pattern theory, offenders may take advantages of criminal opportunities in an awareness space or through communicating with other offenders. Once offenders have a successful experience at some place, the offense at that place could become more routine. Therefore, offenders adapt their routine activity space according to past experience. Routine activity adaptation is implemented using crime templates for offenders and targets. A crime template is defined as a spatial memory of agents. This spatial memory is a grid layer of the simulation area. Each cell of the grid stores the expected reward of crime at the location. As stated earlier in the Q-learning section, the expected rewards of crime stored in the crime template are part of environment rewards for agent spatial learning. According to reinforcement learning theory, a negative reward tends to discourage agents to come to the same place in the future, while a positive reward tends to encourage agents to come to the same place in the future.

The expected rewards stored in a crime template are decided by offender and target past experience. The detailed learning rules for offender expected rewards and target expected costs of crime on crime template are discussed as follows.

Offending template learning

The offending template is the offender's crime template. The expected reward at each crime place is learned by an offender based on past experience. At iteration i+1, if an offender commits an offense, the expected reward at the crime place will be modified according to equation (3.6a):

$$ER_{(x,y)}^{i+1} = ER_{(x,y)}^{i} + M \qquad \ldots\ldots(3.6a)$$

where, $ER^{i}_{(x,y)}$ is the expected reward at iteration i for location (x, y), M is the reward an offender receives following each offense. M is an integer value within [0, 200].

If at iteration i+1, no offense takes place, then the expected reward is subject to temporal decay (which is to mimic the forgetting effect of past experience) as formulated in equation (3.6b):

$$ER_{(x,y)}^{i+1} = (1-C)ER_{(x,y)}^{i} \qquad \ldots\ldots(3.6b)$$

where C is the temporal decay coefficient for offender expected reward. C is a float value within [0, 1].

Following each offense, an offender agent will receive a positive reward (a reward can be understood as the value an offender gets from an offense) from the crime event. This positive reward is then used to update the offending template, using equation (3.6a). The offending template is then used in offender Q-learning, through which the change of offending template is reflected in the cognitive maps. Because offender routine activities are controlled by cognitive maps, offender routine activity patterns will change accordingly. The implications of such spatial adaptations on crime patterns will be examined in chapter five.

The offending template is shared by all offender agents. Therefore, it serves as a way of communication among offender agents. The offending template is like a message board, the knowledge of the environment learned by one offender agent can be shared by all other offender agents. Offender agent communication with each other is a basic and reasonable assumption, since offenders in the real world can communicate with each other through social networks.

(2) Offender motivation development

Following Liang, et. al. (2001), this research assumes that offender agents develop motivation in an "S" curve. An offender agent is initialized to be at the center of the S curve when created. When an offender agent gains more experience (either success or failure), the offender agent will move to the upper tail of the S curve (with a lot of successful experience), or move to the lower tail of the S curve (with a lot of failed experience). An important property of the S-curve learning function is that when an agent gains more experience, the agent tends to learn slower. When an agent has not much experience, the agent learns fast. This is based on the assumption that more adapted offenders are less likely to be changed by short-term experience.

The computational laboratory has a built-in function for offender agent S-curve learning of motivation. The computational laboratory provides an interface for users to control the steepness of the curve. The details of the "S" curve and its effect on crime pattern among offender population will be examined in chapter five.

3.3.2 Target agent

A target agent is derived from a spatial autonomous agent. For a typical crime simulation, a number of target agents need to be created and released to the street network to be robbed by offenders. Besides all the spatial movement and spatial learning capabilities possessed by a spatial autonomous agent, a target agent has necessary properties to attract offender attention, as well as protect themselves.

Target agent properties

Target desirability and guardian capability are two properties of targets in Eck's crime event likelihood evaluation formula. Target desirability can be understood as how much an offender expects to get from the target through crime. Guardian capability is the capability of target self-protecting. Target agents also need to remember past experience for their learning and adaptation. The following properties have been designed for target agents:

Property	Data type	Functionality
(row, column)	Point	Record target agent current position
Cognitive maps	Pointer to grid array	Storing spatial knowledge for routine activities
Activity schedule	Transition matrixes	Scheduling target routine activities
Distance friction	Pointer to a grid	Cost of distance by moving through each cell
Victimizat. template	Pointer to a grid	Spatial memory of victimization experience
Desirability	Numerical	Value an offender expect to get
Guardian capability	Numerical	Self-protection capability
NumVictimization	Numerical	Count of victimization
Kd	Numerical	Slope of desirability adaptation
Kg	Numerical	Slope of guardian capability adaptation

Target adaptation through victimization experience

The feedback assumption by Eck (2003) argued that targets could change guardian capability and desirability following victimizations. The self-policing concept by Bottoms (1994) argued that targets tend to avoid risky areas to avoid victimization. Therefore, this research assumes that target agents change their spatial activity patterns and non-spatial properties following victimizations.

(1) Target spatial activity adaptation

Target routine activity adaptation is implemented using a victimization template. The victimization template is the crime template for target agents. A victimization template is a spatial memory of target agents that records target agents past experience of victimization. The values stored in a victimization template are also expected rewards received by target agents in the past. However, target expected reward is negative because victimization is a cost for targets. Therefore, in this research, target expected reward sometimes is also called expected cost.

Victimization template learning

For target agents, the learning process of expected costs on the victimization template is exactly the same as equation (3.6a), except that the reward following each crime (M) is a negative value for target agents. The temporal decay of target expected cost is exactly the same as equation (3.6b). A target agent will receive a negative reward (cost of crime) following each victimization. This negative reward is then used to update target victimization template using equation (3.6a) (note that M will be a negative value in this case). The Q-learning process of target agents will then reflect the changes of target victimization template into cognitive maps, and change target agents spatial activities accordingly. Target spatial activity adaptation causes the change of routine activities at a given crime place, and changes local situations associated with crime event likelihood at the crime place. The implications of such spatial adaptation on crime patterns and target victimization risks will be examined in chapter five.

The victimization template is shared by all target agents. Therefore, the victimization template is a mechanism for target agent communication with each other. Target agent communication is a reasonable assumption since people in the real world tend to exchange information about crime in many ways, such as through police agencies and television broadcasting.

(2) Target desirability adaptation

Following the RA/CA crime simulation model by Liu, et. al. (2004), this research assumes that targets tend to reduce desirability following victimization. This behavior is modeled by a negative linear function between desirability and victimization count. Users can control the slope of the function, and examine the implications of desirability adaptation on target victimization risks. The details of target desirability adaptation and its implication on victimization patterns will be examined in chapter five.

(3) Target guardian capability adaptation

As a natural reaction to crime, this research assumes that targets tend to increase guardian capability following victimization. This behavior is modeled as a positive linear function between target guardian capability and number of past victimizations. The slope of the linear function can be controlled by users through the interface of the computational laboratory. The details of guardian capability adaptation and its implication on target victimization patterns will be examined in chapter five.

3.3.3 Police agent

A police agent is a spatial autonomous agent without any further development. At the current version of the computational laboratory, police agents are not designed with any adaptation capabilities. Police agents also have no properties that can be used in the crime event likelihood evaluation formula (equation (3.11)). (The lack of adaptability of police agents represents a limitation for the computational laboratory that needs to be researched in the future.) Therefore, police agents do not have spatial adaptation capabilities. Police agents have only one activity in the simulation - executing a routine activity schedule.

The effect of police agents on a crime event is that when a police agent is present at a given place, a crime event will not occur. Police agent properties are listed as follows (which are also the properties of a basic spatial autonomous agent):

Property	Data type	Functionality
(row, column)	Point	Record police agent current position
Cognitive maps	Pointer to grid array	Storing spatial knowledge for routine activities
Activity schedule	Transition matrixes	Scheduling police routine activities
Distance friction	Pointer to a grid	Cost of distance by moving through each cell

3.3.4 Place agent

Place agent refers to a land cell and the manager who is responsible for the management of the place. The collection of all place agents provides the landscape for crime simulation. According to the definition of crime places by Eck (1995), this research set the size of each place at 20 feet.

Place agents have the following properties: management effectiveness and tension. Management effectiveness refers to the effectiveness of a place manager in controlling behaviors at a place. Following the RA/CA model, tension is defined as the overall psychological reaction of place managers to crime events. Crime events, no matter successful or failed, tend to generate psychological reaction such as fear, anxiety, or depression (Hollway & Jefferson 2000; Norris

1997).

The relationship between management effectiveness and tension

Following the RA/CA model, this research assumes that a place manager tends to change place management effectiveness according to local tension level. If tension is high, place managers may intervene in a crime place through enhancing management or increasing surveillance. This relationship is represented as a linear relationship between management effectiveness and tension, as in equation (3.7):

$$\varepsilon^{t}_{(i,j)} = \varepsilon^{0}_{(i,j)} + Km * TS^{t}_{(i,j)} \quad \ldots\ldots(3.7)$$

Where $\varepsilon^{t}_{(i,j)}$ is the management effectiveness at cell (i, j) and iteration t. $\varepsilon^{0}_{(i,j)}$ is the initial management effectiveness. Km is management coefficient. $TS^{t}_{(i,j)}$ is tension at cell (i, j) and iteration t. Km and $\varepsilon^{0}_{(i,j)}$ are subject to user control.

Reaction of crime place to crime event

As stated above, crime events tend to cause psychological tension associated with crime places. Therefore, the reaction of a place agent to a crime event is modeled as tension increase. Following the RA/CA model, at iteration i, the tension at place (*x, y*) is determined by the follows rules:

(1) If there is a crime event occurs at current iteration, then $S^{i+1}_{(x,y)} = S^{i}_{(x,y)} + TI$. Where $S^{i}_{(x,y)}$ is the tension value at previous iteration, and *TI* denotes tension increase following a crime event. $S^{(i+1)}_{(x,y)}$ is tension value at current iteration, which is a positive float value in [0, 100.0];

(2) If there is no crime event at current iteration, then $S^{i+1}_{(x,y)} = S^{i}_{(x,y)} - TD$, with $S^{(i+1)}_{(x,y)} >= 0$. *TD* is temporal decay factor, which is a small positive value (e.g., 1.0). The temporal decay factor is used to represent the “forgetting” effect of tension.

A Cellular Automata model of tension diffuse over space and time

The tension caused by crime events at a crime place tends to affect its neighborhood places, as established by the RA/CA crime simulation model. This neighborhood effect is modeled as the diffusion of tension over crime places. The diffusion of tension is modeled by a CA model. In the following, the CA model for tension diffusion is described.

(1) Neighborhood template

The Moore neighborhood template (each cell having 8 neighboring cells) is used as the neighborhood template of the tension diffusion CA model. The tension of the central cell P_0 is updated according to the tension of its eight neighbors (P_1-P_8). The updating rules are transition rules of the CA model, which are described as follows.

(2) Transition rules

First, the difference between the tension value of the central cell and its eight neighborhood cells at iteration t is calculated as equation (3.8):

$$Sigma = \sum_{k=1, S_k > S_0}^{8} (S_k{}^t - S_0{}^t) \quad \ldots\ldots(3.8)$$

where, *Sigma* is the accumulated difference between S_0^t (place tension value of central cell) and S_k^t (place tension values of neighboring cells) at iteration t. Here, only those S_k^t that is greater than S_0^t is involved in the calculation. This is based on the assumption that people tend to pay more attention to bad news than ordinary news. The tension value of one cell at iteration t+1 can only be affected by those neighborhoods that have higher tension value than itself at iteration t.

A barrier *Beta* is set up for *Sigma*. If Sigma is greater than *Beta*, then there is significant difference between the tension of the central cell and the tension of its neighborhood cells. Then spatial diffusion will occur. The diffusion of tension from neighborhood cells to the central cell is defined in equation (3.9).

If *Sigma* >= *Beta*, significant difference exist. Then

$$S_0^{t+1} = S_0^t + K_{sd} \times Sigma\,,\quad S_0^{t+1} <= 100 \qquad \ldots\ldots(3.9)$$

where, S_0^{t+1} is the place tension value of current cell at iteration t+1. K_{sd} is spatial decay coefficient, which stands for the percent of Sigma diffused to the current cell. It is a float value within [0, 1].

If *Sigma* < *beta*, the difference is not significant, no spatial diffusion will happen.

3.4 Crime event rules

Routine activities bring people together, and the interaction of people generates crime events. Agents for crime simulations are presented in the last section. This section is devoted to model the interaction among offenders, targets, and controllers.

When we formulate the crime event rules, the concept of place manager will be introduced as a controller. Eck (1995) first introduced the concept of place manager in routine activity theory. In this research, it would be appropriate to assume that there is a place manager associated with each place (cell). According to Eck (1995), the role of place manager is to regulate the behavior at a place. Effective management of a place by the place manager should prevent crime at that place. Following the work of Liang et. al. (2001), we denote the effectiveness of place manager in controlling crime as management effectiveness.

Police are reasonable guardians for targets. However, the most important guardians, as argued by environmental criminologists, are ordinary people on the street (Clark & Felson, 1993). People on the street can watch for each other and protect each other. Another type of crime controller is intimate handler of offender. At this stage, intimate handlers are not included in the model. Therefore, three types of controllers are considered in the model: place manager, police and people on the street.

3.4.1 Crime event likelihood

It is reasonable to assume that offender is the decision-maker in a crime event, and the target is always in a passive position. Eck (1995) gave a formulation for evaluating crime event likelihood under a certain situation. This has been reviewed in the literature review section. Formula (2.1) in the literature review section is represented as follows:

$$L(S_{tijk}) = \frac{\delta_{tik} T_{tik} \mu_{tijk} O_{tijk} \alpha_{ti} P_{ti}}{(1+\gamma_{tik} G_{tik})(1+\beta_{tijk} H_{tijk})(1+\varepsilon_{ti} M_{ti})} \quad \ldots\ldots(3.10)$$

In formula (3.10), $L(S_{tijk})$ is the crime likelihood for situation S_{tijk}. The variables T, G, O, H, P and M represent respectively the presence of a target, a guardian, an offender, an intimate handler, and a place. If either of these elements presents in situation S_{tijk}, the variable takes a value of 1. Otherwise they take values of 0. δ, γ, μ, β, α and ε represent respectively target desirability, guardian capability, offender motivation, handler intimacy, place accessibility, and management effectiveness. The values of δ, γ, μ, β, α and ε are within the domain of [0, 1].

There are a few modifications of formula (3.10) need to be made before it can be applied in the simulation model. First, H and β are dropped because handlers are not addressed in the simulation. Second, if crime event likelihood is not zero, then *T*, *O*, and *P* must be 1 meaning that a target, an offender, and a place must be present. Because a guardian (*G*) can be the target, so if a target is present, then a guardian is present. So G is 1. Following the commercial simulation model developed by Liang (2001), we assume that accessibility is put into the concept of management effectiveness (accessibility can be managed by a place manager). With these simplifications, formula (3.10) becomes:

$$L = \frac{\delta\mu}{(1+\varepsilon)(1+\gamma)} \quad \ldots\ldots(3.11)$$

Formula (3.11) is used to evaluate crime event likelihood (a probability value between [0, 1]) in this simulation modeling. The crime event likelihood is positively related to target desirability δ and offender motivation μ, and inversely related to management efficiency ε and guardian capability γ.

3.4.2 Threshold for the occurrence of crime events

Formula (3.11) only gives a probability value for a crime event. In order to decide whether a crime event occurs or not, a uniform random number b_0 is generated within range [0, 1] and compared with L value of formula (3.11). A crime event does not occur when L is equal or less than b_0. A crime event occurs if L is greater than b_0. Because b_0 is a uniform random number within range [0, 1], this way of deciding crime event occurrence guarantees that the probability of a crime event is exactly the likelihood value L. However, for an offender, a crime event can be successful or unsuccessful (Wright and Decker, 1997). To simulate the failure case, a fixed barrier b_1 ($0<=b_1<=1$) is used. Given a crime event occurs, L is then compared with b_1, if L is greater than b_1, the crime event is successful. Conversely, the crime event is unsuccessful if L is not greater than b_1. The value of b_1 can be used to control the chances of crime event failure. If b_1 is 0, then no crime event will fail; if b_1 equals 1.0, then all crime events fail.

3.4.3 Guardian capability γ

Guardian capability refers to the capability of a target or a guardian in protecting the target. There are three types of guardians: police, the target, and people on the street. The effect of police on crime events will be considered separately, here guardian capability only considers the target and people on the street.

From an offender's perspective, the guardian capability (γ) in formula (3.11) is defined in equation (3.12):

$$\gamma_j = \sum_{Ti} \gamma_{Ti} \qquad \ldots\ldots(3.12)$$

Where T_i is any target within i^{th} order neighborhood of target j. Equation (3.12) summarizes the guardian capabilities of all targets within the neighborhood, and use it as guardian capability for a given target. The value of i (the order of the neighborhood, i = 1, 2…) sets a distance about how far away targets can protect each other.

3.4.4 Crime event rules

Based on the crime event likelihood evaluation formula, the crime event rules can be formulated. From an offender decision-making perspective, at each iteration:

(1) If there are no target presents at current place, then L = 0. There are no crime events;

(2) Offender does a vision test within neighborhood (of order v). If there are one or more police agents present, then L=0. No crime event will occur;

(3) If targets are present and there are no police at the current place, then calculate crime event likelihood L using formula 3.11;

(4) Generate a uniform random number b_0 within range [0, 1]. Compare L value with threshold b_0. If L is greater than b_0, then crime event occurs. Otherwise, crime event will not

occur;

(5) If no crime event occurs according to the above random test, then current decision-making iteration ends. If crime event occurs, then L is compared with fixed barrier b_1. If L is greater, then the crime is successful; otherwise, the crime fails.

3.5 Summary - a general spatial crime simulation model

This chapter has presented the conceptual design of the RA/CA/ABM computational laboratory. A type of spatial autonomous agent for routine activity simulation is first introduced, and then agents for crime simulations are introduced, which include offender agent, target agent, and police agent. The grid landscape (street network) for crime simulation is also treated as a set of agents (place agents). Offender agent and target agent are able to perform routine activities according to a routine activity schedule. Eck's (1995) crime event likelihood evaluation formula is modified to provide a set of rules for governing crime event occurrence. Offender agents and target agents are able to change their spatial activity patterns and properties according to past crime experience.

A typical spatial crime simulation involves a number of offender agents, target agents, police agents, and a grid landscape (which is typically a street network). Offender agents, target agents, police agents, and crime places interact on the grid landscape and create crime patterns for analysis. The goal of spatial crime simulation experiments is to explore the implications of agent interactions and adaptations on crime patterns.

Figure 3.14 shows the interaction and feedback relationship among crime event and components of a crime event. Solid lines mean "being applied in", and dotted lines mean "feedback and effect". Solid lines show how different parties contribute to a crime event, while dotted lines show how a crime event feedbacks to affect those parties. For example, the arrow point from “offender motivation” to “crime event rules” shows that offender motivation contributes to the crime event likelihood. The dotted line pointing from “crime event rules” to “offender motivation” means that a crime event effects offender motivation (through the S-curve learning function). Similarly, the dotted line pointing from “crime event rules” to “offender cognitive maps” means that crime events change offender cognitive maps and change offender spatial activities.

A general spatial crime simulation model

This chapter has outlined the necessary components for a crime simulation. If we only provide these necessary components as class modules to users, it would still be difficult to implement crime simulation models, because considerable programming work is needed in order to put these components together and start them to run. This generates another necessary component for the computational laboratory – a general spatial crime simulation framework that holds all these components together. With this simulation framework, researchers can design their own crime simulation experiments through customizing it.

Routine activity theory suggests that a crime event requires the convergence of several elements: an offender, a target, and a place. According to routine activity theory, a standard spatial crime simulation framework can be constructed as illustrated in Figure 3.14. This research assumes that any crime simulation is a special case of Figure 3.14. For example, a crime simulation that does not include police agents (a simulation that contains only offenders, targets and crime places) is a special case of Figure 3.14. The only thing needed to create such a simulation is to set the number of police agents to zero.

This chapter has illustrated the conceptual design of the RA/CA/ABM computational laboratory. Next chapter will discuss the implementation of different types of agents, the grid landscape, and the standard spatial crime simulation framework.

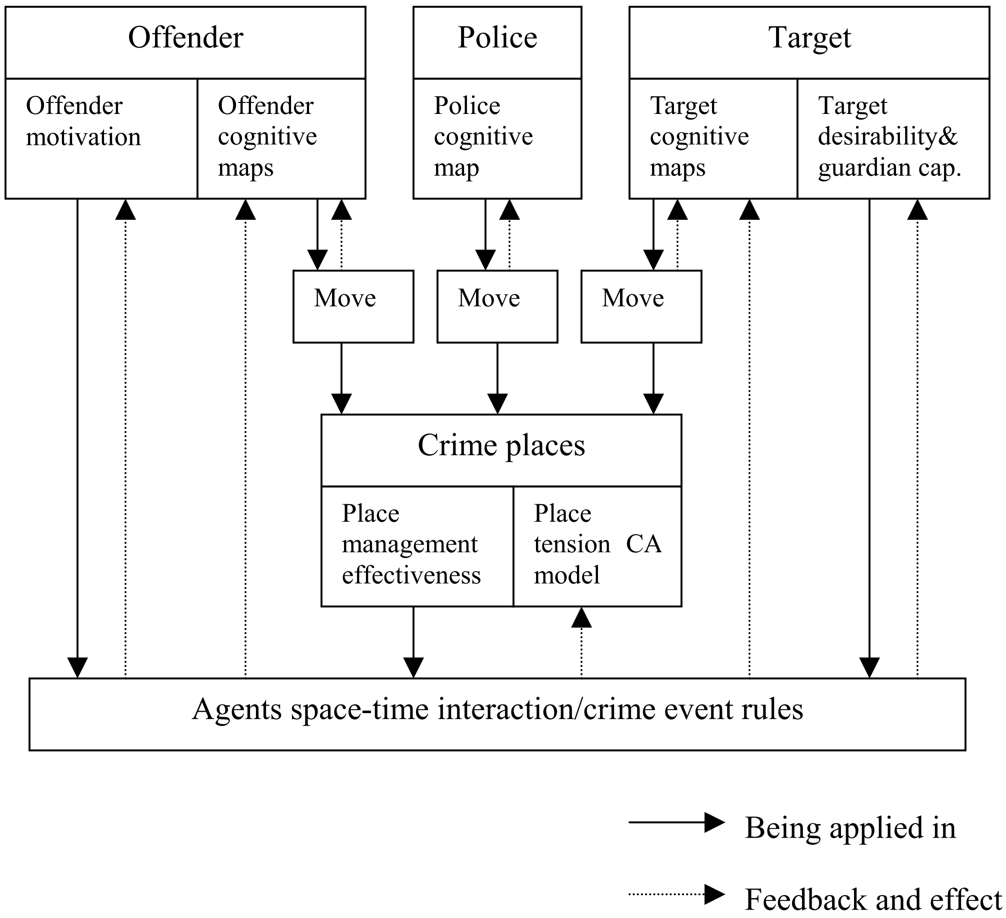


Figure 3.14 The interaction among agents, crime places, and crime events in a typical spatial crime simulation

Chapter 4 Implementation Issues

The purpose of the RA/CA/ABM computational laboratory is to provide an interactive simulation tool to explore the effect of agent interactions and adaptations on crime patterns. Another purpose is for demonstrating crime event theories for educational purposes. These applications require the computational laboratory to be implemented as a computer program with input/output interface so that researchers can use it. The previous chapter has illustrated the conceptual design of the RA/CA/ABM computational laboratory, this chapter is devoted to the implementation design.

4.1 Implementation methodology

4.1.1 Relationship with GIS

As discussed in the last chapter, the methodology of the simulation modeling involves agent-based modeling and cellular automata. These two methods are complementary with GIS in space-time process modeling, since GIS is conventionally configured as a spatial database management system.

The simulation modeling needs spatial data as input, which includes agent routine activity nodes and a street network. Its output crime spatial pattern needs to be analyzed and visualized. Therefore, the spatial data preprocessing, spatial analysis, and visualization functionality of GIS is desirable for such a simulation program. There are potentially three implementation strategies in making use of GIS in the simulation program: (1) developing the program within a GIS software package, such as ArcMap8.x. The advantage of such integration is that the spatial analysis functions can be fully used by the simulation program. This method is not used in this research because of some technical limitations of commercial GIS software. Current GIS software is mainly designed for spatial data management and display, it would be inefficient to use it to develop a dynamic simulation modeling system. Further, users do not have a direct control of the display window in GIS software, and it would be difficult to implement dynamic visualization and animation of the simulation process. (2) Developing spatial analysis and visualization functionalities within the simulation program. This approach is not applicable to this research because too much work would be needed to develop those functionalities. (3) Loosely coupling GIS functionalities with the simulation program. Spatial data can be preprocessed by GIS software first, and then be imported into the simulation program or saved as the format that the simulation software can access. The simulation output can be imported back to the GIS software for analysis. The advantage is that the development effort is minimized, yet the GIS data and functionalities can be used.

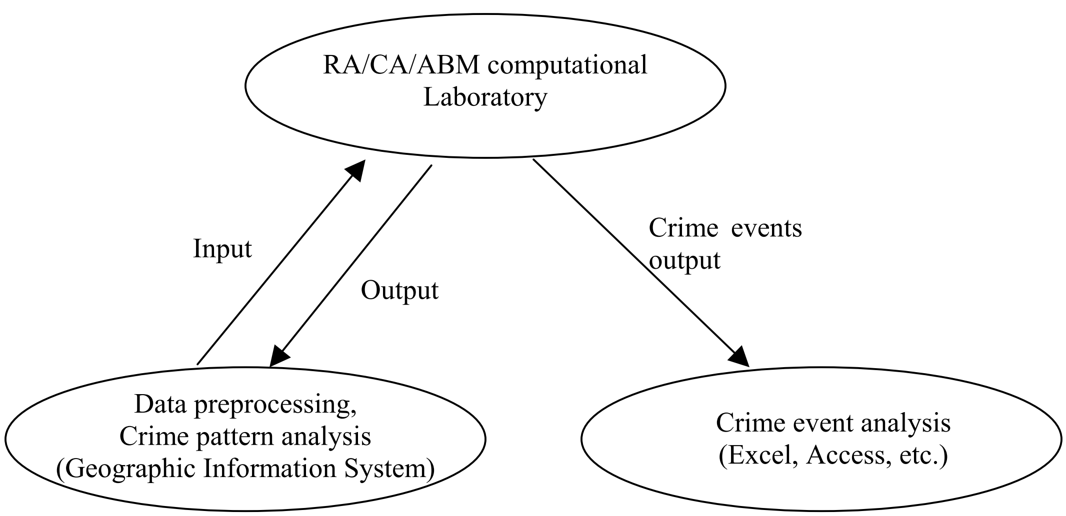


Figure 4.1 Loose coupling of the RA/CA/ABM computational laboratory with GIS

Because of the advantage of the third approach, loose coupling of the simulation program with Arcview3.2 is chosen as the implementation strategy. In order to make use of spatial data provided by Arcview3.2 (mainly grid data and points data), the simulation program has the capability of accessing grid data (.flt files) and texts data (containing a set of (x, y) coordinates) exported from Arcview3.2. The simulated crime patterns and crime events can also be saved as grid data format and text format that can be imported into Arcview3.2 and Excel for further analysis. The relationship between the simulation program and GIS and Excel is illustrated in Figure 4.1.

4.1.2 Implementation methodology

There are many choices for implementing the computational laboratory outside of a GIS. There are standard platforms designed for social scientists to build agent-based simulation models, such as SWARM by the Santa Fe institute, and RePast by University of Chicago. Lower level computer programming language (such as C++) can also be used to develop multiagent system. The advantage of using standard platform to build agent-based simulation model is that developers have a higher starting point. Some existing modules can be used, such as the graphic output. However, the documentation of SWARM (SWARM development group, 2004) indicates that the kind of spatial autonomous agents we needed in crime simulation are not immediately available in SWARM. If we develop the routine activity simulation functions within SWARM, significant efforts are still required. For example, the development of cognitive map, reinforcement learning, and routine activity schedules need to be done with SWARM library. The grid-based object container class in SWARM is also a limitation - each cell only allows one agent to be stored. In crime simulation, many agents can stay in the same cell simultaneously. In this sense, SWARM does not provide an immediate solution for the many functions requited by a crime simulation.

In this research, the RA/CA/ABM computational laboratory is implemented as a standalone program using Visual C++ (which is a type of Object Oriented Programming (OOP) language). OOP languages allow us to implement offenders, targets, and police agents as objects with a set of properties and methods. The properties of objects can be used to represent agent states (such as desirability, etc.), and the methods of objects can be used to implement agent activities (such as spatial movement and committing offense).

Besides the natural similarities between an agent and an object, a number of features of OOP are also desirable for a software project at this size. First, OOP language supports inheritance between objects. A base class can be used to derive many classes and the properties and functions of the base class are inherited. This considerably reduces redundant code. This technique is used in developing offender, target, and police agents based on the base class of a spatial autonomous agent. Second, object-oriented design saves efforts at the design stage because it supports a modular design of the program. The development work can focus on one module each time. Such design makes the development process easier and more manageable.

4.2 Simulation program design

4.2.1 Framework design

At the core of the simulation program is a simulation engine object that contains offender agent objects, target agent objects, police agent objects, and a grid landscape. The simulation engine object is the implementation of the standard spatial crime simulation framework. Environmental criminologists will only need to customize this object to design different simulation experiments. The interrelationship between these class modules are summarized in Figure 4.2.

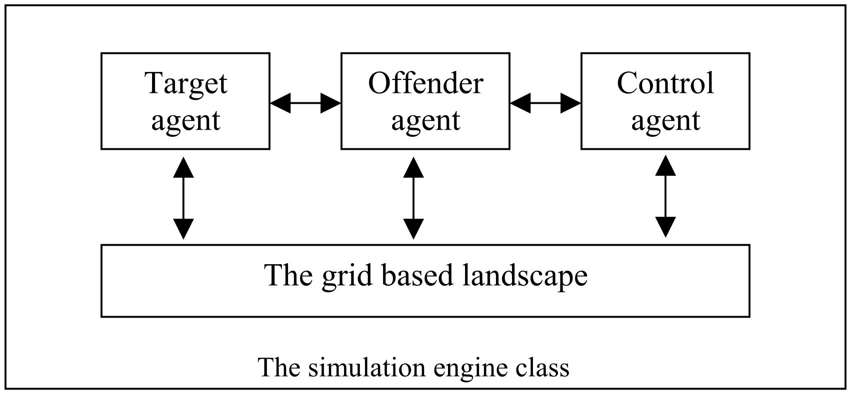


Figure 4.2 The interrelationship between class modules of the simulation program

The simulation engine is wrapped by a user interface for users to customize the simulation according to their needs. The user interface contains a document class (for parameter settings) and a view class (for video display of routine activity patterns and crime patterns). The combination of a user interface with a simulation engine makes the simulation program, as shown in Figure 4.3.

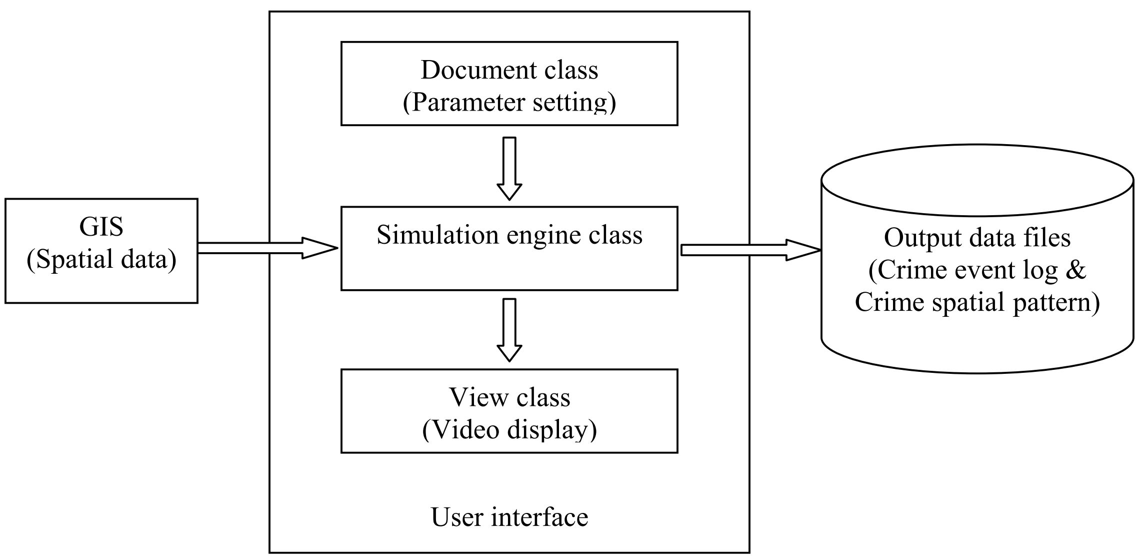


Figure 4.3 The framework of the simulation program

Figure 4.3 depicts the framework and major components of the simulation program. The program starts from the parameter settings document. These parameters are used to initialize the simulation engine. During the initialization process, the simulation engine will read spatial data provided by GIS. After the initialization process is finished, the user can start the simulation. Users can visualize the simulation process through a video display. The crime events and crime patterns generated by the simulation are saved on a local drive for further analysis. This framework can be regarded as the logical design of the RA/CA/ABM computational laboratory.

4.2.2 Class modules design

A few key classes need to be discussed with more details, because these classes implement most concepts that have been presented in the last chapter.

The grid class

Because the simulation program is loosely coupled with GIS, the only communication between GIS and the simulation program is through sharing spatial data. The spatial analysis and visualization capabilities of GIS could not be used in the simulation program directly. The simulation program has to develop its own grid operation class. This class needs the functions of importing data from Arcview Spatial Analyst extension, and some basic local and focal analysis capabilities.

The CGrid class is created to fulfill this requirement, which is capable of reading binary grid/image data exported from Arcview (.flt file), saving the grid data into hard disk, querying cell value, setting cell value, drawing the grid to the screen, as well as overriding some of the operators (e.g., +, -, =) to support simple map algebra.

The CGrid class is used for a number of purposes. First, agent cognitive map and other spatial memories (such as crime templates) are implemented with CGrid as base class. Agent movement is directed by this cognitive map. Also, agents learn and update this cognitive map during movement; second, street network data is saved as a grid and passed to the simulation program as the environment for agent interactions; third, the final crime pattern generated by the simulation is recorded on a grid and saved on hard disk for further analysis; fourth, the cellular automata of place tension spatial diffusion is implemented on a grid.

Agent cognitive map class

Agent cognitive map is the working memory of all types of agents. Agent cognitive map stores the immediate knowledge that an agent needs in order to execute the routine activity schedule. In order to fulfill this need, the agent cognitive map is implemented as CMovementField class.

The cognitive map class contains a grid storing the accumulated least cost values to the destination. The learning functions and agent spatial reasoning (action decision-making) functions are also integrated in the cognitive map class. Thus, the agent cognitive map is an "intelligent" map. By passing a pair of coordinates (row and column number) to the cognitive map object, the cognitive map will tell the agent where to go next step. After the agent takes a step, the cognitive map will take care of the learning process and update itself automatically.

The spatial autonomous agent base class

Offender agents, target agents, and control agents are all agents in the simulation. They share many common characteristics. For example, they all move and learn on the grid with similar rules and cognitive maps. Because this research does not distinguish legal routine activities and illegal routine activities, the routine activity model is the same for offender agent, target agents and control agents.

This program created CAgent as a spatial autonomous agent base class for offender agents, target agents and control agents. The CAgent base class includes agent cognitive maps and routine activity schedules as properties, and a walking function as a method that allows an agent to execute the routine activity schedule.

Characteristics that distinguish offender, target, and control agents are then implemented in derived classes respectively. The difference between an offender agent and a target agent is that an offender agent receives positive reward from crime and uses it to update its offending template, while a target agent receives negative reward from a crime and uses it to update its victimization template; offender agents initiate crimes, while targets are passive; offender agents have motivation as state, while target agents have desirability and guardian capability as states. Control agents do not have any specific states or rewards designed at this stage of research.

Because of these specific characteristics, offender agents are implemented with state transition rules for motivation (as will be discussed in next chapter), and decision-making capabilities for offending. Target agents are implemented with state transition rules for desirability and guardian capability. Offender agents and target agents all learn from experience of crime and adapt spatial activity patterns. Control agents only have the most basic capabilities as defined in agent base class (e.g., executing routine activities).

The routine activity schedule class

Routine activity schedule specifies an agent's activities within a certain period of time (e.g., one day). The routine activity schedule contains several transition matrixes for a one-day period, as defined in the last chapter. Physically, a transition matrix exists as a text file on a local drive. It allows users to define new schedules or modify existing schedules using a text editor. In the simulation program, routine activity schedule is a property of agent, which is implemented as a separate class – CActivityTemplate class.

The functions of the routine activity schedule class include: a two-dimension array of float values storing the activity schedule; and a decision-making function about next activity for an agent, given current activity as input.

Agent dynamic spatial index class

An agent (e.g., an offender in criminal decision-making) sometimes needs to reference other agents by spatial relationship. For example, when an offender agent makes a decision on crime, the offender agent needs to check which target agents are at neighboring cells (because these neighboring agents may provide protection to target agents). In this case, the offender agent needs to search neighboring cells for other targets or police.

One way of doing this is to get the neighboring cells coordinates, and loop through all agents to find whose coordinates match the neighboring cell coordinates. If the coordinates match, then those agents are considered to be in the neighboring cells. The experience in developing the RA/CA/ABM computational laboratory indicates that this kind of neighborhood search is rather time-consuming and is not the best choice.

To facilitate agent referencing through spatial relationship, this research developed a spatial index for agents when agents move on the street network. The simplest spatial index would be a grid layer, with each cell of the grid stores agent ID. Thus, if an agent needs to check which agents are at neighboring cells, the agent only needs to check what IDs are stored in neighboring cells. This agent then can use these IDs to reference those neighboring agents easily.

Each time an agent moves, the agent's ID will be removed from the original position and put to the new position. Because agents are moving, their IDs will also be moving on the index grid. This is why it is called a dynamic spatial index.

Another problem associated with the simple one-layer spatial index is that there may be more then one agent staying in a cell. If the spatial index is only a one-layer grid, then only one agent's ID can be recorded at a given place. Other agents' IDs would not be able to be stored after the place is occupied by an agent. To solve the problem, multi-layer spatial index is used, so that many agents' IDs can be recorded.

The dynamic spatial index has been implemented in the class CSpatialArena. The application of this simple technology in the simulation program is tested to be effective in improving the performance of agent spatial activities. The current implementation of the dynamic spatial index class can handle 30 agents at each place.

The tension class

Tension is the state variable of crime places/place managers. It is an environment state. Last chapter established that tension at a given place will diffuse over neighborhood space and affect other places. This diffusion process is modeled by a cellular automata model.

The CPlaceTension class is created to implement the cellular automata model of tension diffusion. The class has a grid representation of the tension surface, and a function that contains the transition rules of the cellular automata model. The class also provides an interface for querying and setting the tension values on the tension surface.

The simulation engine class

Finally, a simulation engine class (CSimulationEngine) is created to assemble the instances of the above classes. The simulation engine class implements a standard spatial crime simulation framework as summarized in chapter three, which aims to provide a default simulation model for researchers to customize in order to design experiments. The simulation engine class is instantiated in the view object of the simulation program interface, and then can be launched by the user. A document class and several dialogue classes are created to customize the simulation engine. The simulation will generate crime events and crime patterns that are saved in the crime event log class, and can be saved onto a local drive.

The simulation engine mainly contains a number of offender agents, target agents, control agents, a street network, a place tension diffusion model, an agent dynamic spatial index object, a crime event log manager, and a simulation loop controller. The simulation engine has two main functions: one is to initialize the simulation engine, the other is to launch the simulation engine. The initialization parameters come from the document object, and the dynamic distribution and movement of agents are visualized through the view object (the display is a mapping of the agent spatial dynamic index to the screen).

4.2.3 Object model diagrams

The inheritance and containment relationships between the various classes are illustrated by object model diagrams, as in Figure 4.4 and Figure 4.5. Figure 4.4 depicts the classes that are contained in the simulation engine class; Figure 4.5 depicts the components of the agent base class.

In the object model diagram of the simulation engine, offender agents, target agents and controller agents are members of the simulation engine class. Other members of the simulation engine class include a place tension object (the environment), a dynamic spatial index object, a loop controller object, and a crime event log object. The super class of offender agents, target agents and controller agents is CAgent, whose structure is illustrated in Figure 4.5. CAgent class mainly contains a cognitive map (CMovementField) and a routine activity schedule (CActivityTemplate). Each of these classes has their own structure as shown on the diagram.

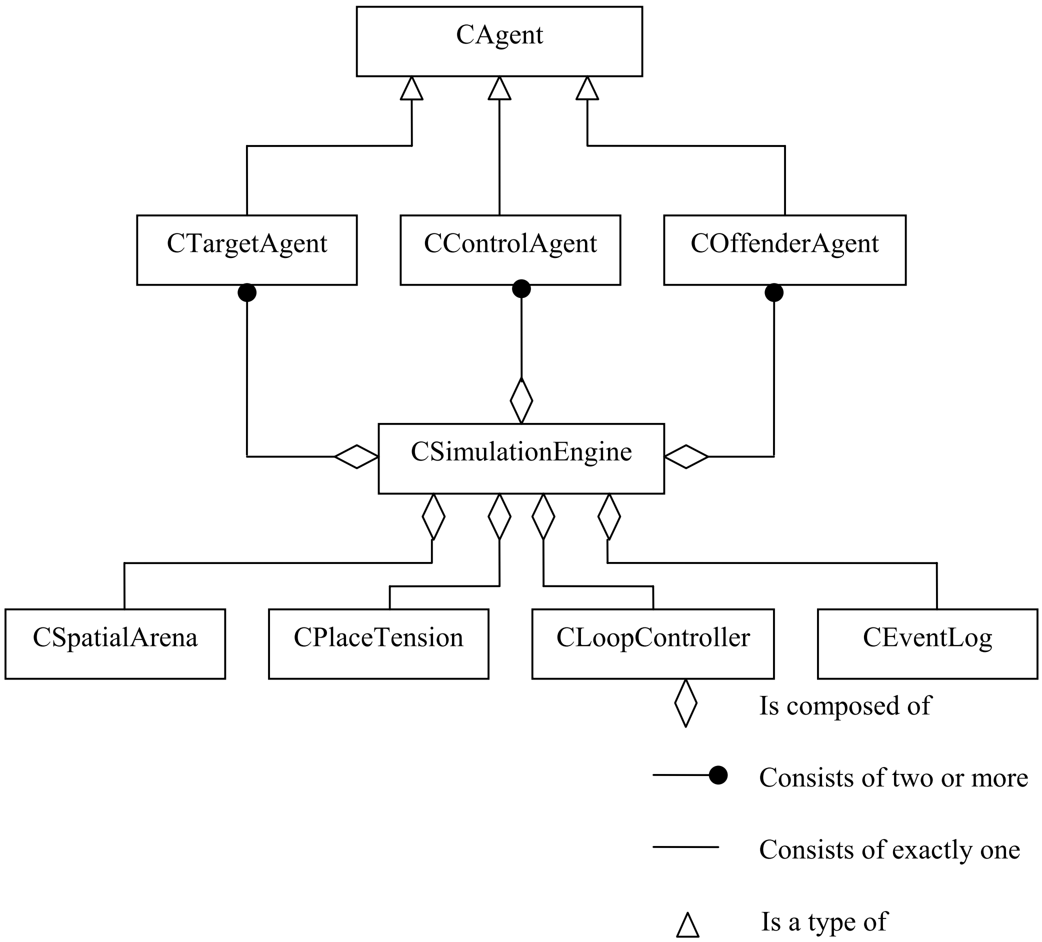


Figure 4.4 Object model diagram of the simulation engine class

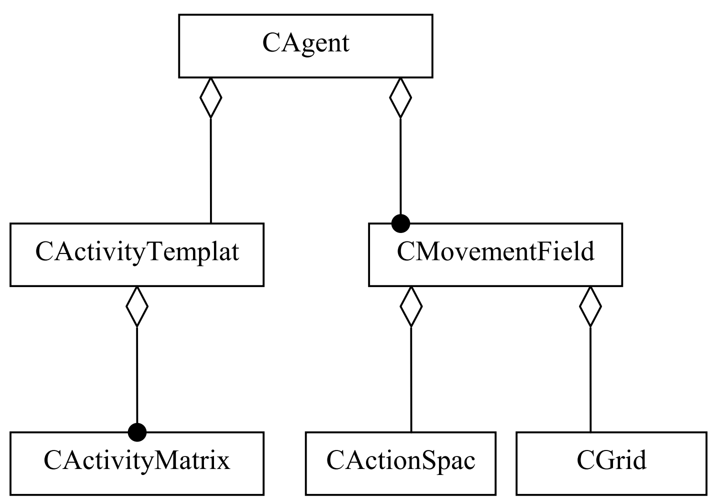


Figure 4.5 Object model diagram of the base class Cagent

4.2.4 User interface design

The simulation program is developed using Visual C++ based on the above framework design and class modules design. The simulation program requires some input parameters and generates crime events and crime patterns as outputs. The input parameters include agent behavioral rules and other settings, such as the number of each type of agents and agent routine activity schedules. All the input parameters are saved as a simulation file that can be opened through user interface. Thus, the users do not need to create a new set of parameters for each simulation.

A. Parameter dialogues

The input parameters are managed by two dialogues. When users open a simulation, create a new simulation, the main parameter dialogue will show up, which allows users to review or change parameter settings. The main parameter dialogue is shown in Figure 4.6.

The main parameter dialogue manages parameter settings that are not changeable during a simulation. For example, the number of offender agents, target agents, and police agents; street network and routine activity nodes; number of simulation iterations.

Another parameter setting dialogue, agent properties dialogue can be accessed through the main parameter dialogue. Agent properties dialogue allows users to review and set agent properties, such as guardian capability coefficient, desirability coefficient, agent routine activity schedules, place management coefficient, tension CA parameters, spatial behavior parameters and crime template learning parameters. The agent properties dialogue can be opened either when the user creates a new simulation, opens an existing simulation, or when a simulation is running. It is particularly useful when the user wants to update agent parameter settings when the simulation is running. The agent properties dialogue is shown in Figure 4.8.

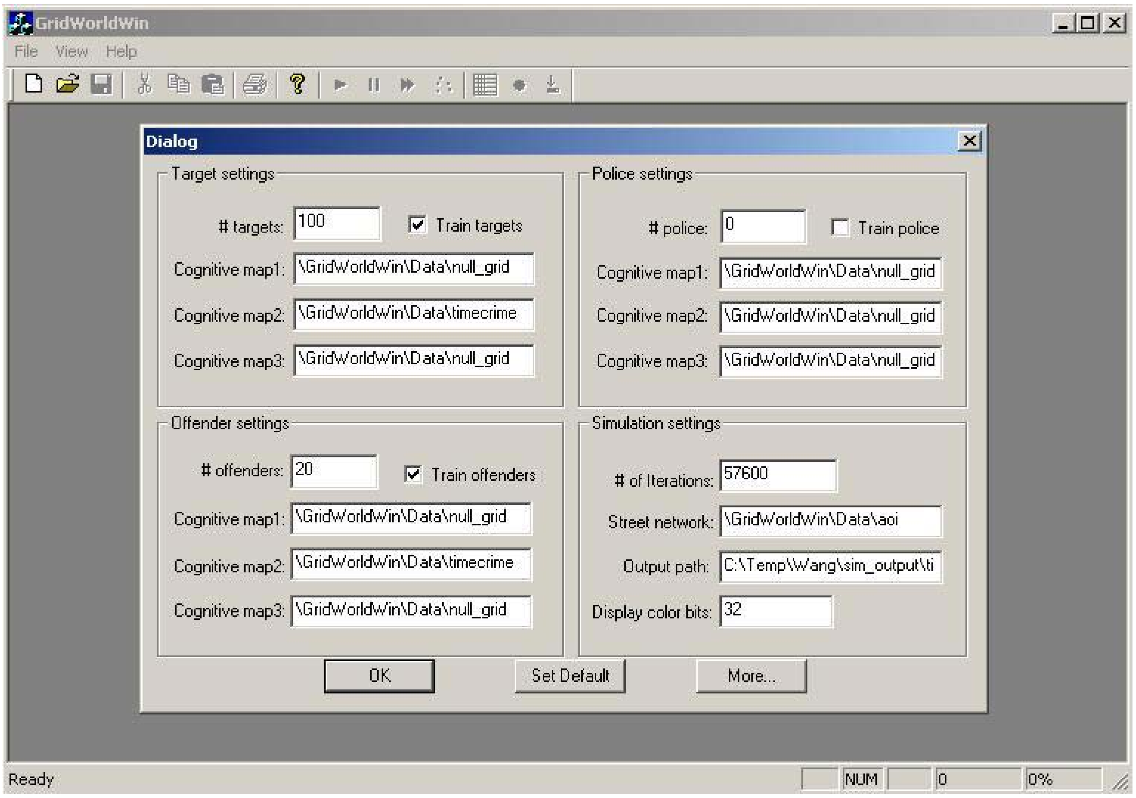


Figure 4.6 The main simulation parameter setting dialogue

B. The video-display window

In order to provide a more intuitive view of a simulation process, a simple dynamic visualization technology is used - a video-display window is created for a simulation. There are two display modes for the video-display window. First, the video-display window displays the movement of offender agents, target agents, and police agents. Second, the same window also displays the crime event spatial distributions. There are buttons on the toolbar that allow users to switch between the two display modes, so that users can observe the co-evolution of routine activity patterns and crime patterns during the simulation. For an example of such video-display window, see Figure 4.7

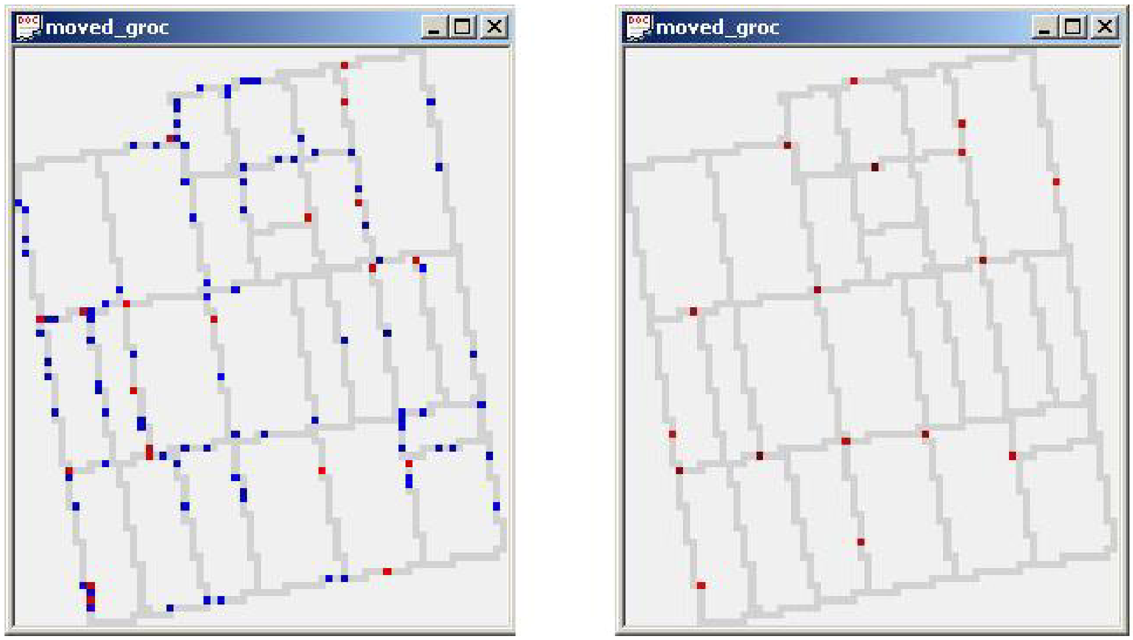


Figure 4.7 The video-display window for agent movement (left, red dots are offender agents and blue dots are target agents) and crime event spatial distribution (right)

C. The simulation control interface

When a user confirms the setting of parameters in Figure 4.6, the simulation is ready to run. A toolbar and a status bar are created to control and monitor the simulation process. Several buttons are created on the toolbar, including starting a simulation, pausing a simulation, fast running mode, switch to display crime pattern, and showing the properties dialogue, etc. A typical simulation interface that integrates a video-display window, the mainframe window (which contains the toolbar and status bar), and the properties dialogue is shown in Figure 4.8.

The elements of the interface are labeled from 1 to 11 in Figure 4.8. Their functions are explained as follows:

1 – Button for create new simulation. After this button is clicked, the main parameter dialogue (as in Figure 4.6) will show up and allows users to specify parameters for a new simulation.

2 – Button for opening an existing simulation setting file.

3 – Save a simulation (the parameter setting file).

4 – Start a simulation. When a simulation is in a paused or stopped status, clicking this button will start the simulation; if the simulation is running is a fast mode, click this button will set the simulation in a normal speed, and show the movement of agents.

5 – Pause the simulation. When a simulation is running (whether in normal speed or in fast mode), clicking this button will pause the simulation.

6 - Speed-up the simulation. A normal simulation contains a video-display window and allows the user to see the simulation process. In order to visualize the simulation process, the simulation is set to be very slow. Clicking this button will disable the display and therefore set the simulation in a fast mode.

7 – Show crime pattern. Normally, the video-display window displays the movement of agents. By clicking this button, the window will be switched to display spatial crime pattern, which is also dynamically refreshed.

8 – Button for bringing up the agent properties dialogue. This dialogue is to review and change agent properties during runtime.

9 – Button for bringing up a dialogue for setting break point for the simulation. The breakpoint allows the simulation automatically pause after a number of iterations as specified by users. This function is especially useful if users want to observe the temporal development of agent properties in a fixed time interval. For example, users can set the simulation to automatically pause after every 100 iterations.

10 – Query agent properties and crime patterns, and save data on a local drive.

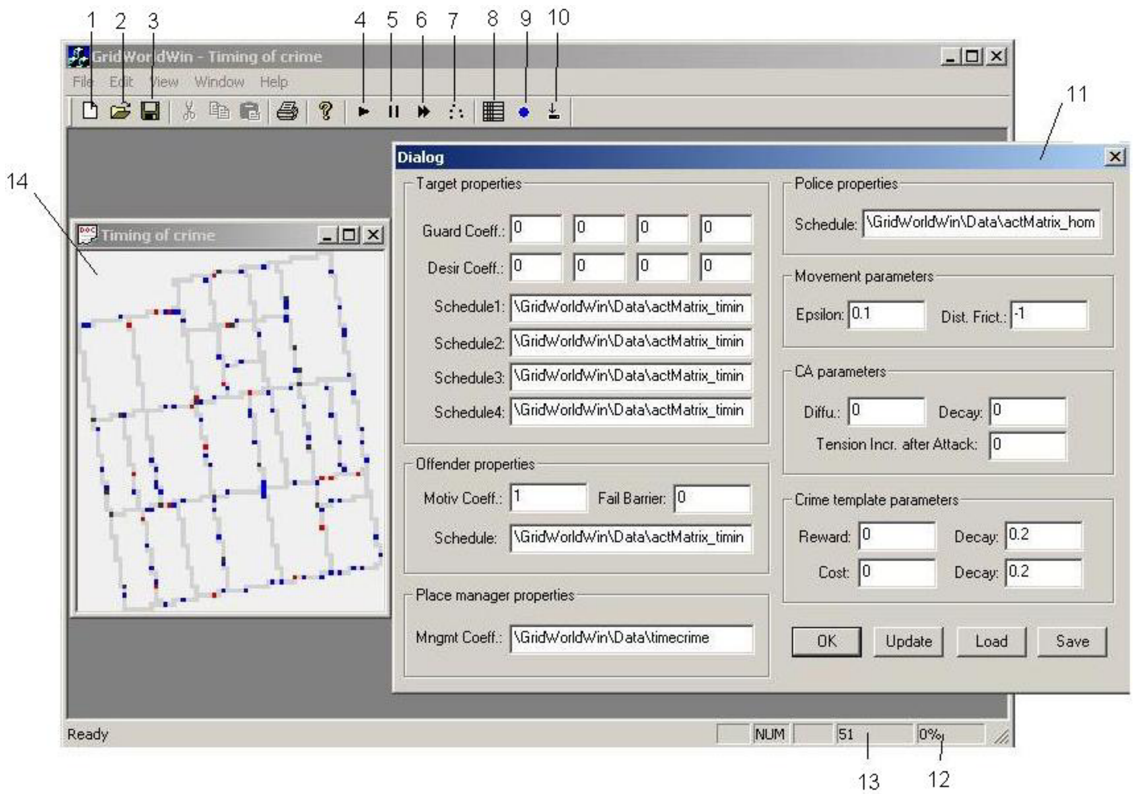


Figure 4.8 The simulation program interface

11 – The agent properties dialogue, for reviewing and updating agent properties during runtime.

12 – Progress indicator (displaying percentage of finished iterations).

13 – Current iteration indicator (displaying current iteration number).

14 – The video display window for showing agent movement animation as well as crime patterns.

4.3 Simulation input and observations

4.3.1 Simulation input and process

In order to run a simulation, researchers need to provide the following information to the simulation program: number of iterations desired for the simulation; number of the three types agents (offender agents, target agents, and police agents); a street network; routine activity schedules for each of the three types of agents (unless the number of one type is 0); routine activity nodes for each type of agents corresponding to the routine activity schedules; a management effectiveness layer; desirability, guardian capability, and motivation settings; the cellular automata parameters; crime template parameters; etc. For a detailed introduction to parameter setting, see chapter five and appendix A.

After a simulation is initialized with parameters provided by users, the simulation engine can be launched by clicking button 4 as discussed in the last section. This button will call the "Run" function of the simulation engine. The main simulation loop is defined in the "Run" function. The "Run" function runs a loop at each iteration. The total number of iterations is defined by the user and is stored in a simulation loop controller object. The most basic activity of each agent is a step of movement in a single iteration. The "Run" function of the simulation engine puts all agents into movement in space and time.

For every agent, there are other activities associated with movement. For example, crime decision-making and state transitions of offender agent are accompanied with each step of movement. Those activities are implemented in the "Move" function of offender agent, target agent and police agent.

For offender agent, the movement function includes the following activities: spatial movement; spatial learning; crime decision-making; state transitions following crime event (e.g., change of motivation value).

For target agent, the movement function includes the following activities: spatial movement; spatial learning; state transitions following victimization (change of guardian capability and desirability).

If an offender agent commits a crime, then the crime event will be stored in a crime event log object, and then saved in a log file on a local drive. The simulation starts at iteration zero, and ends when the specified number of iterations is finished.

4.3.2 Simulation observations

An important goal of the simulation program is to support interactive simulation. Users need to monitor the simulation process, query agent properties during runtime, and analyze simulation results. The simulation program is able to generate several types of observations for a crime simulation experiment: a dynamic visualization window, a crime event log, and a crime spatial pattern grid. The crime event log and crime spatial pattern are automatically saved on a local drive when the simulation finishes. Users need to specify the paths and names for the crime event log and crime spatial pattern file on a local drive.

However, for some simulation experiments, the temporal development process of crime patterns and agent states are of concern to researchers. For example, the temporal development of offender motivation is useful in studying the repeat offending problem, so the user needs to record the temporal process of motivation development. The simulation program is designed with the functionality of querying agent states during runtime and saving the result on disk.

Crime event log and crime spatial pattern grid

The agent-based crime pattern simulation output files include a crime event log file and a crime spatial pattern file. The crime event log file (in ASCII format) is designed to record every detail of a crime event, including the start time of the simulation, sequence ID numbers of the crime events, the iteration number when the crime event occurs, day of the crime event, hour of the day, row and column number of the crime event location, offender agent ID and target agent

ID. With this log file, the user can analyze non-spatial pattern of crime occurrence, such as the offending pattern, victimization pattern, and temporal trend of crime.

A grid file is used to record the spatial distribution of crime events. Each cell of the grid file stores the number of crime events at that location. At the end of the simulation, the grid file will be written onto a local drive.

Query agent states and crime patterns during runtime

When a simulation runs to a point where we want to query the states of agents and the development status of the crime spatial pattern, we can temporarily pause the program (either manually or through setting up breakpoints), and then save the current states of agents (such as offender motivation) and the spatial pattern of crime (a grid file) onto a local drive. This can be done using button 10 illustrated in Figure 4.8.

Chapter 5 Simulation experiments with the RA/CA/ABM computational laboratory

The computational laboratory provides a tool to facilitate the development of crime place theories based on routine activity theory. The full potential of computation laboratories for environmental criminology is far beyond a dissertation research. This research focuses on theoretical experimentation within the computational laboratory for research purposes, and theory demonstration within the computational laboratory for teaching purposes.

As illustrated in chapter four, the RA/CA/ABM computational laboratory is targeted for the Microsoft Windows desktop, with a standard Windows program interface. With the computational laboratory, one can easily design and finish an experiment within a couple of hours. Within the computational laboratory, different crime patterns can be observed by manipulating the parameters and agent routine activity schedules. The simulation can be run again and again, and the results can be analyzed by statistical methods so that we know how robust the pattern is. Another advantage of the computational laboratory is that experiments are nondestructive in a simulation environment, in contrast with real world experiments.

In order to validate the model, demonstrate the procedures of designing experiments with the model, and show the usefulness of the model, this chapter is devoted to simulating some spatial, temporal, victimization, and offending crime patterns. Some of the patterns are well established in empirical research literature, and these patterns are used to validate the model by showing that the model is actually able to produce credible crime patterns. Another purpose of simulating these well-known crime patterns is to assist setting up parameters for the model. With a video display interface, such simulations are able to demonstrate crime event and crime pattern theories for teaching purposes. Other crime patterns simulated in this chapter are less well researched in literature, and these are used to show that the model is able to generate new hypotheses to be tested against real world evidence.

In the following sections, how to design experiments and set up parameters with the RA/CA/ABM computational laboratory is discussed first, and the crime pattern simulation experiments follow. Even though police agents are part of the computational laboratory, the functions associated with police agents still need more research. Therefore, in this chapter, police agents have not been included in any experiments.

5.1 Experiment design procedures and parameter settings

Designing an experiment with the simulation program involves specifying environment backcloth settings, such as number of each type of agents, street network, routine activity nodes, and CA parameters, and agent property settings, such as agent routine activity schedules, movement parameters, crime template parameters, and agent state adaptation parameters.

In the following, an example of a simulation scenario is used to demonstrate how to configure the computational laboratory for a certain simulation purpose (a more detailed parameter setting procedure is presented in Appendix A). This scenario is designed to test the spatial adaptability of offender agents and target agents. There are four cases for this scenario: offender agents with low adaptability and target agents with low adaptability; offender agents with low adaptability and target agents with high adaptability; offender agents with high adaptability and target agents with low adaptability; offender agents with high adaptability and target agents with high adaptability. In this scenario, we are interested in observing the dynamic space-time distribution of agents and the evolution of agent routine activity patterns.

5.1.1 Street network and routine activity nodes

The street network is the space where agents perform routine activities and interact with each other. Routine activity nodes are destination places for agent routine activities. Examples of routine activity nodes include shops, work places, homes, schools, etc. The first step of designing the experiment is to specify the street network and routine activity nodes for offender agents and target agents.

The computational laboratory provides a function for creating a new simulation. For the interface of creating new simulation, see Figure 4.6 and Figure 4.8. When button "1" in Figure 4.8 is hit, the main parameter setting dialogue will show up as in Figure 4.6. In Figure 4.6, users can set up street network and routine activity nodes for offender agents and target agents.

There are many possible ways to configure the street network and activity nodes. The simulation area can be very large (such as a whole urban area), as well as very small (such as a few blocks). In this research, because the crime events and crime patterns we are concerned are at micro place level (which is sub-neighborhood resolution), a few neighborhoods or blocks should be enough for a theoretical simulation purpose. In order to test the adaptabilities of offender agents and target agents, an urban area within downtown Cincinnati was chosen. This area is between 14th St and Central Parkway from North to South and between Race St and Clay St from West to East, containing 25 blocks, as shown in Figure 5.1. A certain number of offender agents and target agents will be released onto the street network to perform some routine activity schedules. Three types of routine activity nodes are created for all agents: a bus stop, two shops, and a work place. The street network is converted to grid data using Arcview GIS, and then exported into exchangeable data format that can be accessed by the computational laboratory.

The paths and names of street network and routine activity nodes are put in the main parameter dialogue of Figure 4.6.

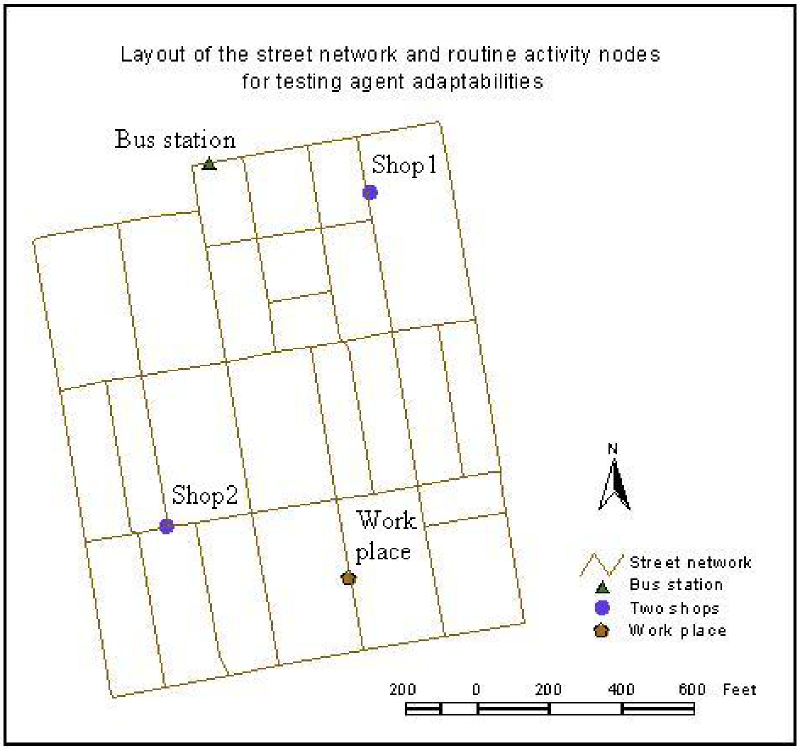


Figure 5.1 Street network and routine activity nodes for the scenario of testing offender agents and target agents adaptabilities

5.1.2 Agent property parameters

After the street network and routine activity nodes are specified, the next step is to set up agent property parameters. Agent properties include: number of each type of agents, routine activity schedules for each agent, distance friction, and crime template parameters.

Number of agents

In this scenario, the number of offender agents is set to 20, and the number of target agents is set to 100. The number of police agents is set to 0. (In order to save computation resources, the upper bound of the number of offender agents has been set to 100, and upper bound of the number of target agents has been set to 300 in the current version of the computational laboratory. However, this limitation can be modified and improved through changing source code of the software.)

Routine activity schedules

Target routine activities are set as the movement from the bus station, to a shop and to the work place. The sequence of target movement from bus station to shop to work place is fixed. The transition probabilities of target activities are defined as Table 5.1.

Table 5.1 Transition probabilities of target activities

Activity1\Activity2	**Bus station**	**Shop**	**Work place**
Bus station	0	1.0	0
Shop	0	0	1.0
Work place	1.0	0	0

Offender routine activities are a combination of random movement and shopping. The shopping frequency is defined as 0.01 as in Table 5.2. Random movement is a special type of activity designed for offender agents. When an offender agent is in random movement, the offender agent randomly decides the next cell to step into. Therefore, there is no destination for an agent when the agent is in random movement status. Different from other activities, when an agent is in random movement status, the agent may change activity status every iteration according to the transitional probability (while the agent is involved in other types of activities, the agent changes activity status only when the destination of current activity is reached). The transition probability for shopping is 0.01, meaning that offender agents go shopping once every 100 iterations. After offender agents finish the shopping, they return to random walking, according to Table 5.2.

Table 5.2 Transition probabilities of offender activities

Activity1\Activity2	**Shop**	**Random Movement**
Shop	0	1.0
Random Movement	0.01	0.99

Distance friction

According to basic geographic theories, all human activities are subject to distance friction. As presented in chapter three, distance friction is one of the spatial reward factors in an agent's learning of the cognitive map, which determines the paths and destinations of agent routine activities. Distance friction factor affects the mobility and adaptability of agents.

In this scenario, the distance friction is set to be –1. If under such a distance friction setting, agent adaptability is too high, then distance friction should be increased; on the contrary, if agent adaptability is too low, then distance friction should be decreased.

Crime template parameters

Crime template parameters should directly determine the adaptability of offender agents and target agents. As defined in chapter three, crime templates store the expected cost (for targets) or reward (for offenders) values at different places. For the offending template, there are two parameters: reward increment following each crime, and a temporal decay coefficient of the expected reward. For the victimization template, there are also two parameters: cost increment following each crime, and a temporal decay coefficient of the expected cost.

Each of these parameters affects agent adaptability. For example, when reward increment is set to be high, then the offender is more sensitive to crime, and is more easily to be attracted to places with higher crime rates. Similarly, when cost increment is set to be high, then the target is more sensitive to victimization experience. The temporal decay coefficients affect how long an agent can remember past experience. For example, if target cost temporal decay is high, then targets tend to forget past experience and pay more attention to current experiences.

Initially, these four parameters are set to be 0, so that offender agents and target agents have no adaptability. These four parameters will be adjusted later to see how it affects agent adaptability.

5.1.3 Scenario one: agent spatial learning and adaptation

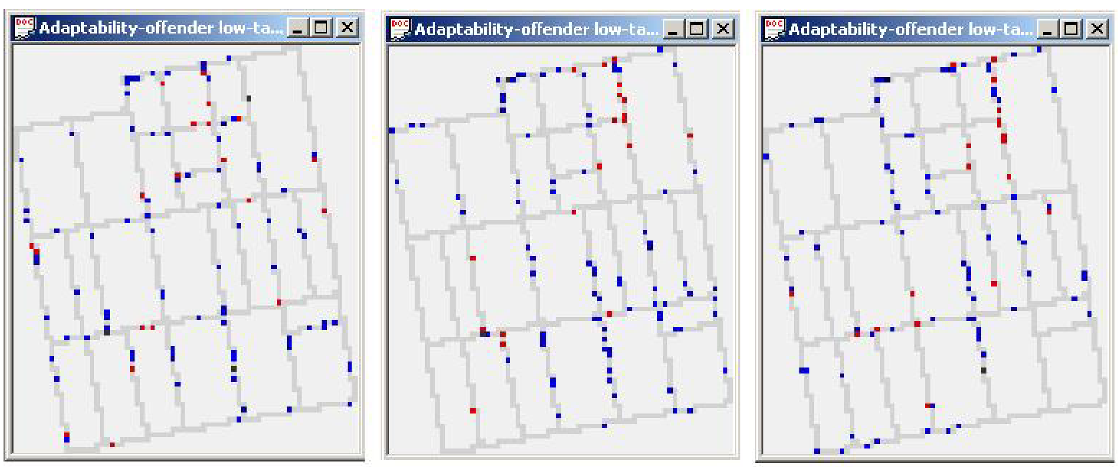
This scenario is designed based on the parameter settings of 5.1.1 and 5.1.2 to test whether offender agents and target agents possess the learning and adaptation capabilities as designed in chapter three.

Routine activity nodes and street network of agent activities are set up as Figure 5.1. Target agents are scheduled to move from the bus station to a shop to the work place, and loop. Offender agents are scheduled to "hang around" each of the two shops (their routine activities are scheduled as a combination of random movement and shopping). Below, four cases are listed, each with different combination of offender adaptability and target adaptability.

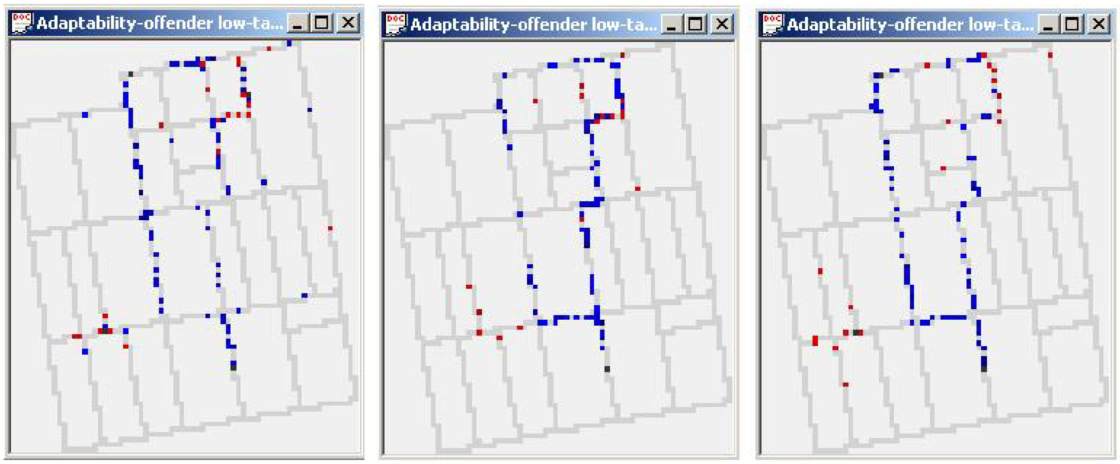
Case 1: Offender with low adaptability, target with low adaptability. Offender reward is 0, target cost is 0, and the temporal decay coefficients for offender reward and target cost are all set to 0.

In this case, how offenders and targets learn their routine activity space will be shown. When offender agents and target agents are initialized, their positions are random. They also have no spatial knowledge. But agents will gain spatial knowledge through a large number of trials. Offender agents learn to shop at the two shops and they start to gather around the two shops after some time. Target agents learn a near-shortest path from the bus station to a shop and to the work place and then return to the bus station. This process is shown in Figure 5.2. Offender agents and target agents finally develop a stable and equilibrium activity pattern, as shown in panel (f) of Figure 5.2. At this moment, offender motivation (μ) is set to be 0, so that there are no crime events.

Starting from iteration# 1500, μ is set to be 3.0, so that crime events could occur when offender agents and target agents meet. Up to iteration# 3000, several street robberies have already occurred around shop 1, as shown in Figure 5.3 panel (a). However, the movement pattern of offender agents and target agents still remains unchanged (as shown in panel (b) of Figure 5.3). This indicates that when offender reward and target cost are set to zero, they do not receive feedback from environment, therefore, both the offender population and the target population have very low adaptability.

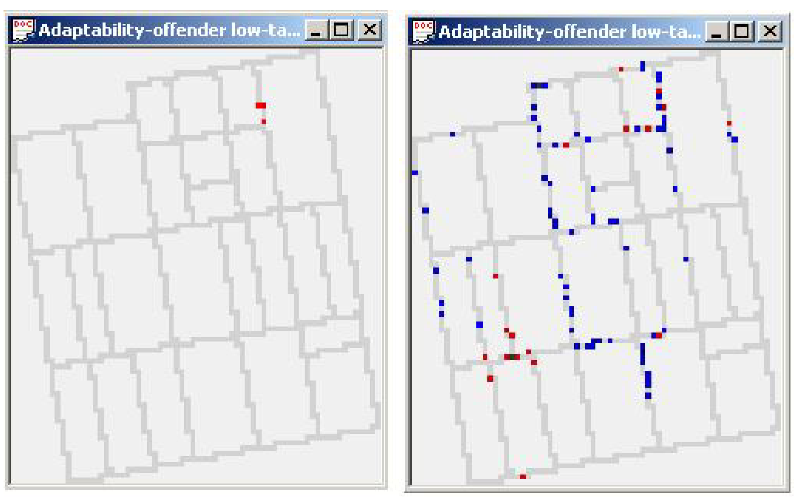


(a) Iteration# = 15, μ=0 (b) Iteration# = 300, μ=0 (c) Iteration# = 600, μ=0



(d) Iteration# = 900, μ=0   (e) Iteration# = 1200, μ=0   (f) Iteration# = 1500, μ=0

Figure 5.2 Offenders and targets learning of activity space in case 1 (red dots are offender agents, blue dots are target agents)



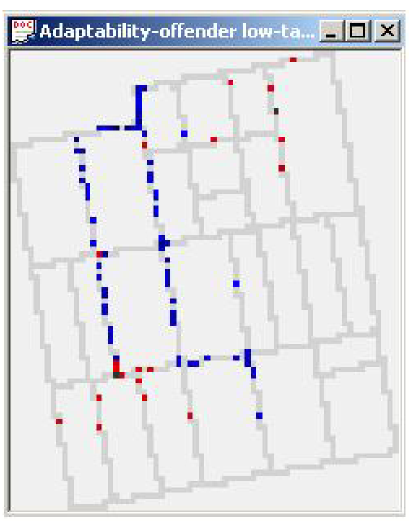
(a) Crime events near shop1   (b) Movement patterns of agents

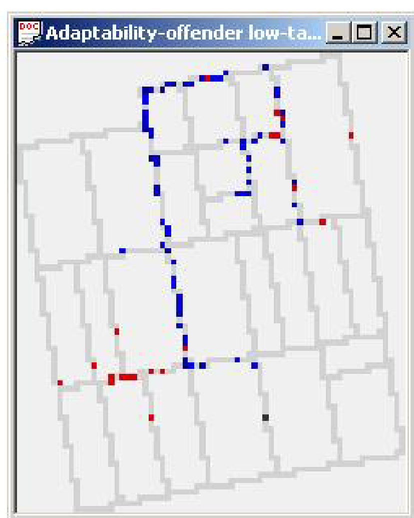
Figure 5.3 The distribution of crime events and agent activity patterns at iteration# 3000

Case 2: Offender with low adaptability, target with high adaptability. Offender reward is 0, target cost is 200, offender reward temporal decay coefficient is set to 0, and target cost temporal decay coefficient is set to 0.2.

This case is to show that target agents are able to switch their shopping locations when crime events occur. Again, at iteration# 0, offender motivation is set to 0. At iteration# 2000, agents develop activity patterns similar as Figure 5.2 panel (f).

At iteration# 2000, offender motivation is set to be 3.0. Because target agents are attacked at shop 1, they switched their shopping location to shop 2 at iteration# 2470, as shown in Figure 5.4 panel (a). However, because crime events occur at shop 2, and targets tend to forget past victimization experience at shop 1 (which is controlled by the target cost temporal decay coefficient), target agents are driven back to shop 1 again at iteration# 3100, as shown in panel (b) of Figure 5.4.





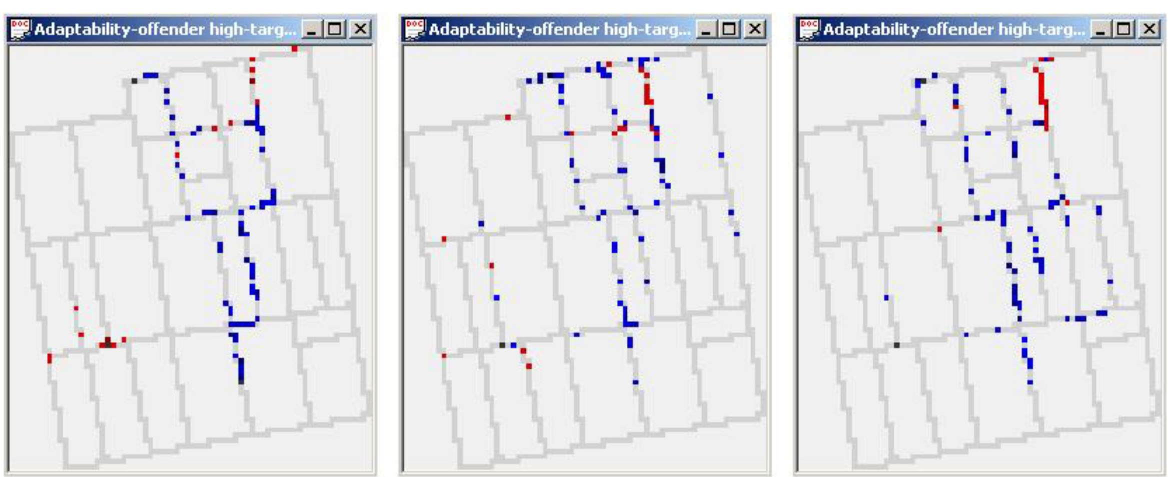
(a) Agent distribution at iteration#2470 (b) Agent distribution at iteration# 3100

Figure 5.4 The adaptation of target agents in case 2

Case 3: Offender with high adaptability, target with low adaptability. Offender reward is 200, target cost is 0, offender reward temporal decay coefficient is set to 0.2, and target cost temporal decay coefficient is set to 0.

This case is used to test whether offender agents have adaptability or not. Before iteration# 2000, offender motivation is set to 0. There are no crime events before iteration# 2000, and offender agents and target agents develop activity pattern similar as Figure 5.2 panel (f). Offender agents hang around the two shops.

At iteration# 2000, offender motivation is set to be a constant 3.0. After iteration# 2000, crime events start to occur. However, criminal opportunities mainly concentrate at shop 1, since target agents pass through it. Then at iteration# 2300, offenders start to move to shop 1, because of the attraction of criminal opportunities there. At about iteration# 2500, offender agents have all switched to shop 1, as shown in Figure 5.5. This shows that when offender agents receive reward from crime and communicate with each other, they are adaptable.



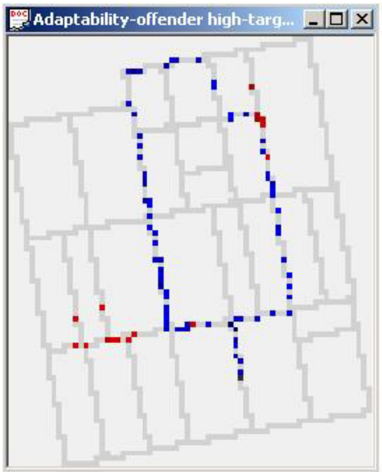
(a) Iteration# 2000 (b) Iteration# 2300 (c) Iteration# 2500

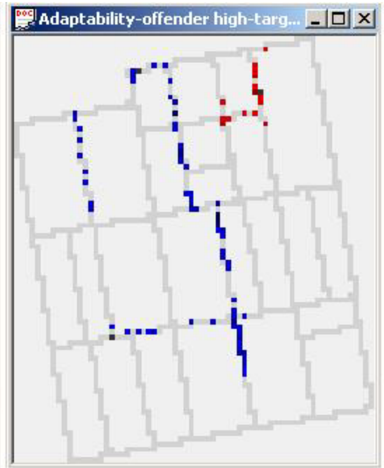
Figure 5.5 The adaptation process of offender agents in case 3

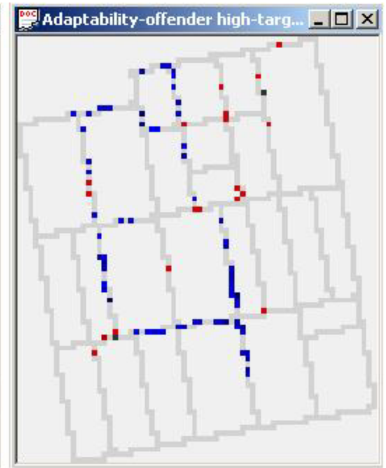
Case 4: Offender with high adaptability, target with high adaptability. Offender reward is 200, target cost is 200, offender reward temporal decay coefficient is set to 0.9, and target cost temporal decay coefficient is set to 0.9.

Case 2 and Case 3 tested that offender agents and target agents show adaptability when they receive reward (or cost) from crime events. This case is to show what will happen when both offenders and targets possess adaptability.

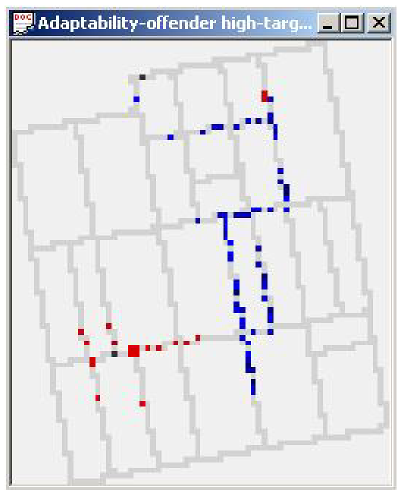
Before iteration# 2000, offender motivation is set to be 0, so that there are no crime events before iteration# 2000. Up to iteration# 2000, the activity pattern developed by offender agents and target agents is shown in panel (a) of Figure 5.6. At iteration# 2000, offender motivation is set to 3.0. The adaptation process after iteration# 2000 is shown in Figure 5.6. The general trend that can be seen from Figure 5.6 is that the target population and the offender population tend to be separated from each other (see panel (b) and (d) of Figure 5.6), because targets aim to avoid victimization. However, offender agents tend to catch up with target agents (see panel (c) of Figure 5.6), because they are attracted by the criminal opportunities. Once offender agents catch up, the amount of time offender agents and target agents stay together is short, and target agents adapt again (see panel (d) of Figure 5.6). In this case, the process that target agents fleet risky area to avoid victimization and offender agents catch up to get criminal opportunity, will oscillates without stabilizing in an equilibrium status.







(a) Iteration# 2000 (b) Iteration# 2300 (c) Iteration# 3000



(d) Iteration# 3400

Figure 5.6 The adaptation process of offender agents and target agents when their adaptabilities are both high

5.2 Scenario two: timing of crime

In this section, a simulation scenario is designed to test the hypothesis that temporal patterns of crime in a society are structured by routine activity rhythms of everyday life. In this scenario, the computational laboratory can be regarded as an artificial society (for definition, see Epstein, 1996), in which a population of offender agents and target agents are released onto a street network with a number of routine activity nodes. These agents then are scheduled to perform some routine activities on the street network, and crime events are observed and analyzed.

In this scenario, 22 shops are sparsely distributed on the same street network as scenario one, as shown in Figure 5.7. 100 targets and 20 offenders are released on the street network. The routine activities of these targets and offenders are a combination of random movement and shopping.

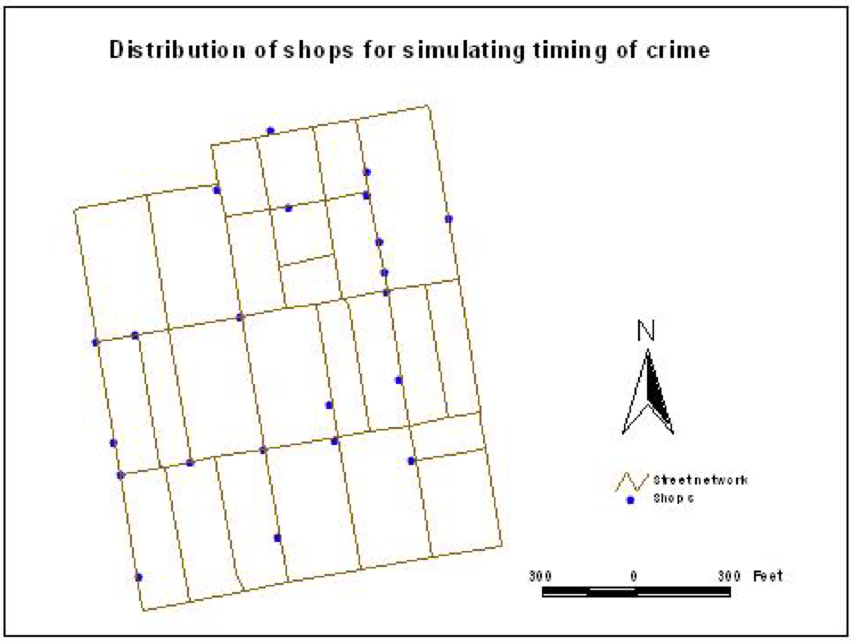


Figure 5.7 The layout of street network and shops for scenario two

The management effectiveness is set to 0.5 at the shop locations, and 1.0 at other places. Crime events are more likely to occur at shop locations. To test whether timing of crime follows activity rhythms of offenders and targets, the 24 hours of a day are divided into eight intervals. The transition probabilities of random movement to shopping are specified to be different among the eight time intervals. The goal is to see if the numbers of crime events among the eight time intervals follow the transition probabilities of legal activities. As expected, more shopping activities produce more chances for offenders and targets to meet at places with low place management effectiveness, therefore, more crime events are expected at time intervals with high shopping probabilities.

Table 5.3 Transition probabilities (from random movement to shopping) for both offenders and targets in Scenario two

Time of the day	Interval#	Trans. Prob.
00:00-3:00	0	0.005
3:00-6:00	1	0.005
6:00-9:00	2	0.05
9:00-12:00	3	0.005
12:00-15:00	4	0.05
15:00-18:00	5	0.005
18:00-21:00	6	0.05
21:00-00:00	7	0.005

Table 5.3 listed the transition probabilities of offenders and targets activities (from random movement to shopping). Three time intervals of the day are set to have high probabilities of shopping, which are 6:00-9:00, 12:00-15:00, 18:00-21:00. Eight transition matrixes corresponding to the eight time intervals are developed as input for the computational laboratory.

Offender motivation is set to 1.0 and fixed, while target desirability and guardian capability is set to 0.5 and fixed (the change of these properties will be studied in later scenarios). Under these settings, the simulation runs for 57,600 iterations (10 days). The crime events are recorded in a log file. Over the 10-day simulation period, crime events are summarized into 24 hours of a day, which is listed in Table 5.4.

Table 5.4 Summary of crime events in 24 hours over the 10-day simulation period

Hour	Crime count	Hour	Crime count	Hour	Crime count
0	39	8	94	16	65
1	46	9	44	17	43
2	43	10	32	18	99
3	48	11	48	19	95
4	50	12	93	20	81
5	53	13	88	21	63
6	77	14	87	22	45
7	104	15	38	23	45

From Table 5.4, time intervals 2 (including hours 6, 7 and 8), 4 (including hours 12, 13 and 14) and 6 (including hours 6, 7 and 8) have more crime events. In Table 5.3, these three intervals have higher transitional probabilities, i.e., higher shopping frequencies for both offenders and targets. This pattern is confirmed in Figure 5.8, where the temporal distribution of crime is shown to have three peaks at the three intervals with higher transitional probabilities. Panel (b) of Figure 5.8 shows the peaks of transitional probabilities that correspond to the peaks of the crime rate in panel (a). Routine activity theory argues that temporal trends of crime can be explained by changes of lifestyles or routine activities in the society. This simple scenario shows that the computational laboratory can simulate temporal crime patterns that conform to known crime theories.

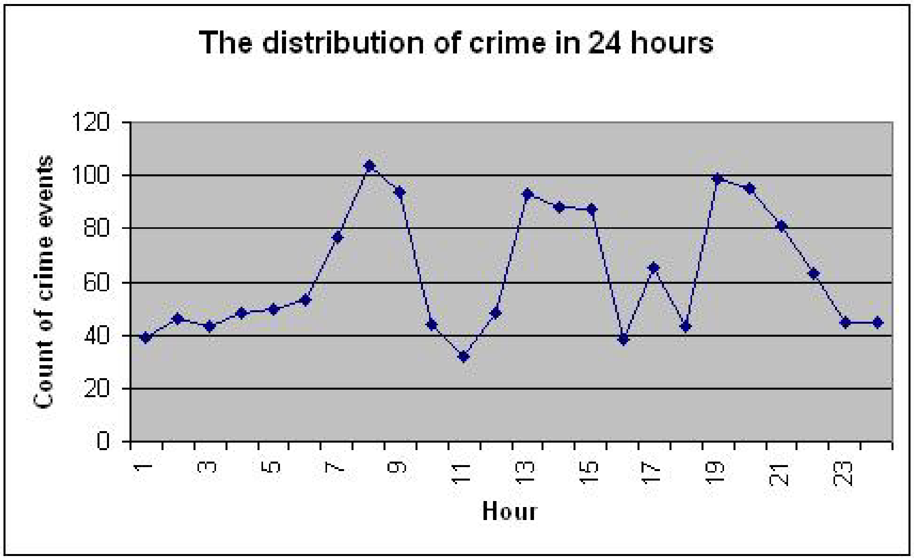


Figure 5.8 (a) The distribution of crime events in 24 hours

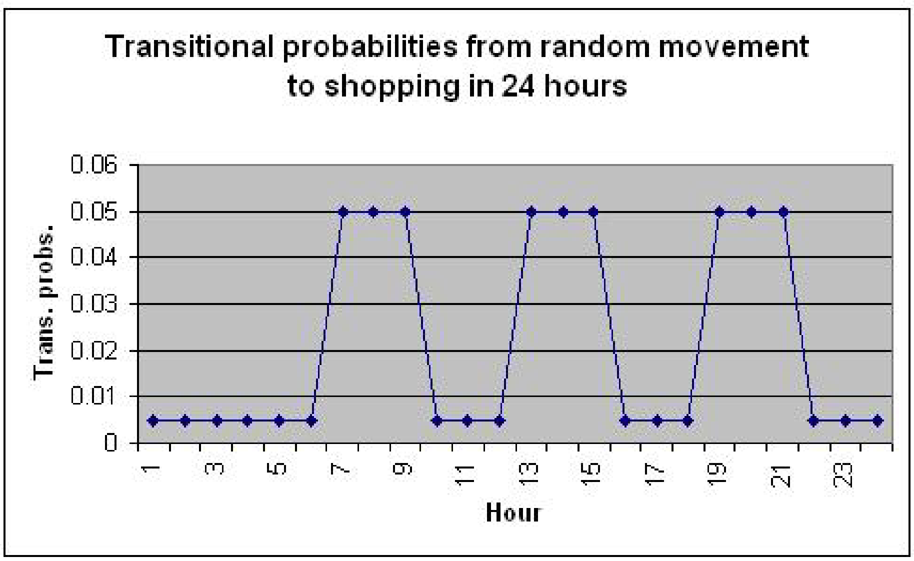


Figure 5.8 (b) Transitional probabilities (from random movement to shopping) in 24 hours of a day

Figure 5.8 Peaks of crime rate (a) and peaks of shopping frequencies (b)

5.3 Scenario three: Repeat location

Environmental criminologists have established that the spatial distribution of crime is not random. In fact, at the micro crime place level, the distribution of crime is highly skewed. For example, the top 10 percent addresses with most crime events usually account for about 60 percent of total crime (Eck, 2001). The fact that a small percentage of crime places accounts for disproportionate high percentage of crime events is known as repeat locations, or repeat places, or the "dens of iniquity" problem in environmental criminological literature (Eck, 2001). The highly skewed distribution of crime events among crime places is shown to fit a power function by Spelman (1995).

Repeat location, as a spatial characteristic of crime patterns, offers an opportunity to test our simulation model. If the model is correct, we expect that repeat locations can be observed from the simulated crime pattern. The process of testing our simulation model also shows if the empirical regularity of repeat locations is the outcome of agents interacting within the theoretical contexts of routine activity theory and crime pattern theory. In the following, several cases are examined to study the spatial characteristics of crime generated by the simulation model.

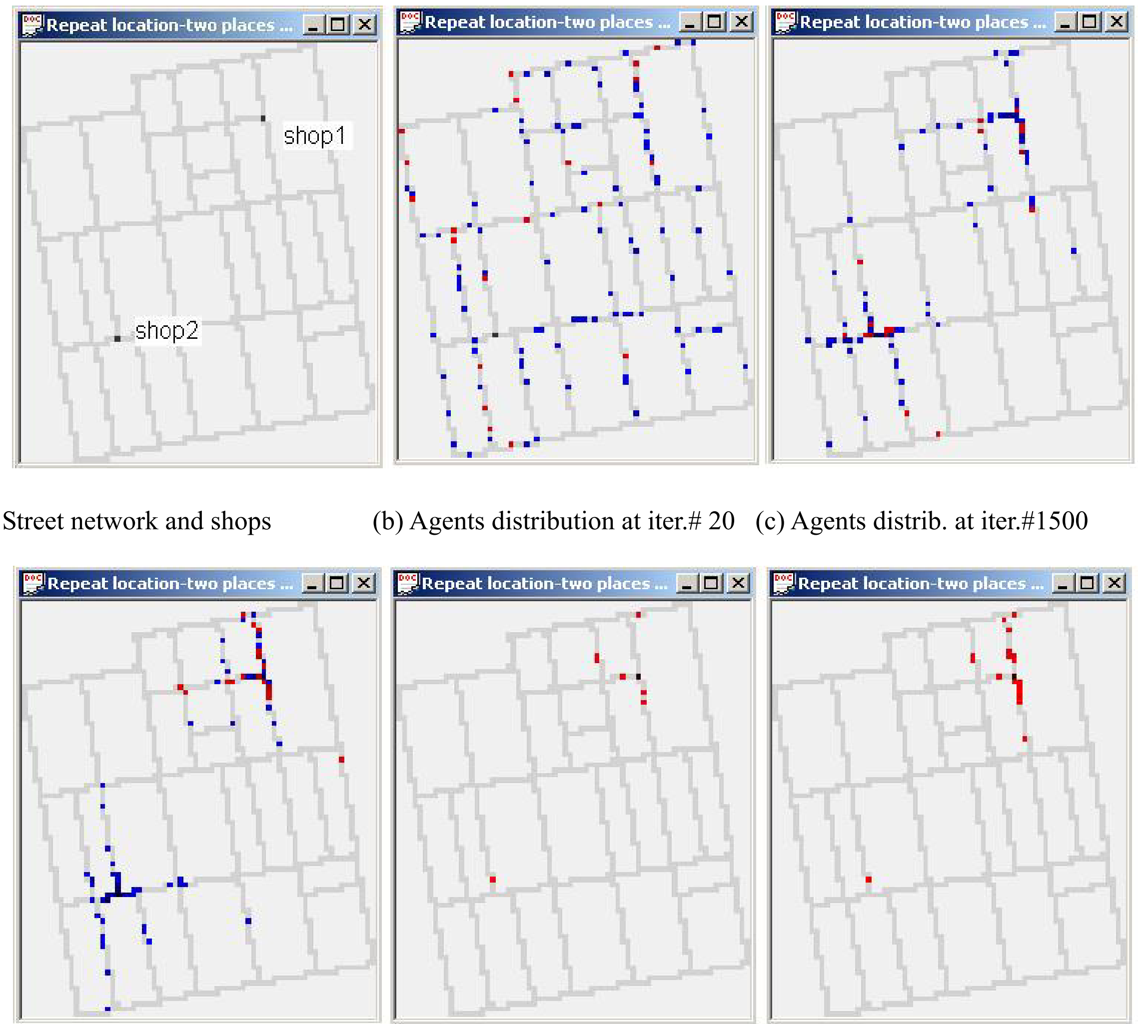
Case 1: Two crime places with non-equivalent management effectiveness

In this case, two shops are created for offender agents and target agents as activity nodes. The street network is the same as scenario two. Agents (offenders and targets) are set with two types of activities on the street network: random movement and shopping in the two shops. The transitional probabilities from random movement to shopping are set to 0.01 for all agents. Offender reward following each crime is 100; target cost following each crime is also set to 100 (In other words, offender agents and target agents are all set to be adaptable). Figure 5.9 panel (a) illustrates the locations of the two shops and the layout of street network.

The management effectiveness at the two shops are set to be different. At shop 1, the management effectiveness is set to be 0.1, and at shop 2, the management effectiveness is set to be 1.0. Therefore, shop 1 is managed less effectively. According to the management effectiveness concept established by Eck (1995), shop 1 is more crime-prone.

Eck (2003) explained that the skewed distribution of crime events among crime places is caused by the malfunction of controllers (in this case, the controller is place manager). When an offender commit an offense at a place, if the place manager fails to respond to the crime, then the offender may get reinforced and commit more crime at the same place. This simple case with only two shops is able to test this explanation of crime event concentration. We expect to see that offenders discover less effectively managed places and commit offenses routinely at those places.

The outcome of the simulation is illustrated in Figure 5.9. Before iteration# 1500, offender motivation is set to be 0, and agent spatial distribution is shown in panel (c). In panel (c), there are about half offender agents and half target agents distributed around both shops. In panel (d) and (e), the simulation goes to iteration#2000, and the distribution of crime events and distribution of offender and target agents show some interesting patterns. From panel (e), it is obvious that there are more crime events at and around shop 1 in comparison to shop 2. Offender agents learned that there is more reward for crime at shop 1, so they were attracted to shop 1. This is why almost all offender agents are drawn to shop 1 (panel (d)). On the other hand, target agents tend to avoid crime, so most of them left shop 1 and gathered around shop 2. A small number of target agents remained at shop 1. Recall chapter three, where target agents are designed to be able to guard each other, so the fleeting away of target agents leaves shop 1 with insufficient guardianship. Thus, shop 1 is a situation with a lot of offenders, a small number of target agents, and a lot of crime events. This is the process that enables shop 1 to develop into a crime hot spot, while shop 2 remains a cold spot. This pattern becomes stable without external intervention. In iteration# 3000, more crime events occurred around shop 1 (see panel (f)), and the difference between the two shops is even larger. Table 5.5 lists some of the statistics of panel (c), (d), and (e). The results generated in this simulation conform to expectations of Eck (2003).



(a) Street network and shops (b) Agents distribution at iter.# 20 (c) Agents distrib. at iter.#1500

(d) Agents distrib. at iter.#2000 (e) Crime distrib. at #iter.#2000 (f) Crime distrib. at iter.#3000

Figure 5.9 Simulation process of scenario three case 1

Table 5.5 Number of agents and crime events associated with shop 1 and shop 2

Shop	Iter.#1500		Iter.#2000	
	Number of agents (offender/target)	Number of crime events	Number of agents (offender/target)	Number of crime events
Shop 1	11/53	0	20/31	32
Shop 2	9/47	0	0/69	3

<u>Repeating the simulation for more times</u>

The above result is based on a single simulation. Because there are random errors with the model, a natural question is whether the patterns observed are robust or not. If we repeat the simulation, will we get the same spatial distribution every time? Or, how often could we observe shop 1 becoming a hotspot?

To address these questions, the same simulation was run for 10 times. For 8 times, we observed that shop 1 became a hotspot, and for the other 2 times shop 2 became a hotspot. Recall that shop 1 has a lower management effectiveness value than shop 2. Therefore, we can predict that shop 1 is more likely to become the hotspot (with 80% chances). This case shows that stable hotspots are a function of poor management.

Case 2: Two crime places with equivalent management effectiveness

In case 1, the skewed distribution of crime is explained by the difference of management effectiveness. One might expect that if the management effectiveness is equal among the two places, then the two shops will have an equal number of crime events associated with them. Is this true? Or in other words, will the distribution of crime events tend to be skewed even without the condition of non-equivalent management effectiveness? This case sets out to examine this problem.

In this case, all parameter settings are the same as case 1, except that the management effectiveness are equal for shop 1 and shop 2 (both 0.1 in this case). Again, offender motivation is set to be 0 before iteration# 1500, and there is no crime event before iteration # 1500. The simulation generates similar distribution of agents as in panel (c) of Figure 5.9.

After iteration# 1500, offender motivation value is set to 3.0. It turns out that the distribution of crime events between the two places is skewed despite the fact that the two sites have the same management effectiveness. One outcome of the simulation under this case is shown in Figure 5.10. The outcome illustrated in Figure 5.10 shows that shop2 develops into a hot spot of crime (with much more crime than shop 1). At iteration # 2200, there are 15 offenders and only about 34 targets gathering around shop 2. Around shop 1 location, there are 66 targets and 5 offenders gathered. Around shop 2, there are 61 crimes occurred till iteration# 2200, while around shop 1, there are only 10 crime occurred in the same period. Just as in case 1, the relative large number of offenders and small number of targets at shop 2 form a positive feedback to the spatial pattern of crime – more offenders bring more chances of crime at shop 2, while less targets reduce the guardianship at shop 2. Thus, the skewed distribution of crime is able to keep when the simulation goes on. For example, panel (c) of Figure 5.10 shows that shop 2 is still much hotter than shop 1 when the simulation runs to 4000 iterations.

(a) Agents distrib. at iter.#2200 (b) Crime distrib. at iter.#2200 (c) Crime distrib. at iter.#4000

Figure 5.10 One simulation outcome under the condition of equivalent management effectiveness for the two shops

For the simulation outcome represented in Figure 5.10, the development processes of crime rates (defined as number of crime events over every 100 iterations) at shop 1 and shop 2 are illustrated in Figure 5.11. As shown in Figure 5.11, starting from an equal crime rate (4 incidents per 100 iterations), shop 2 becomes more and more crime-prone, while shop 1 gradually cools down.

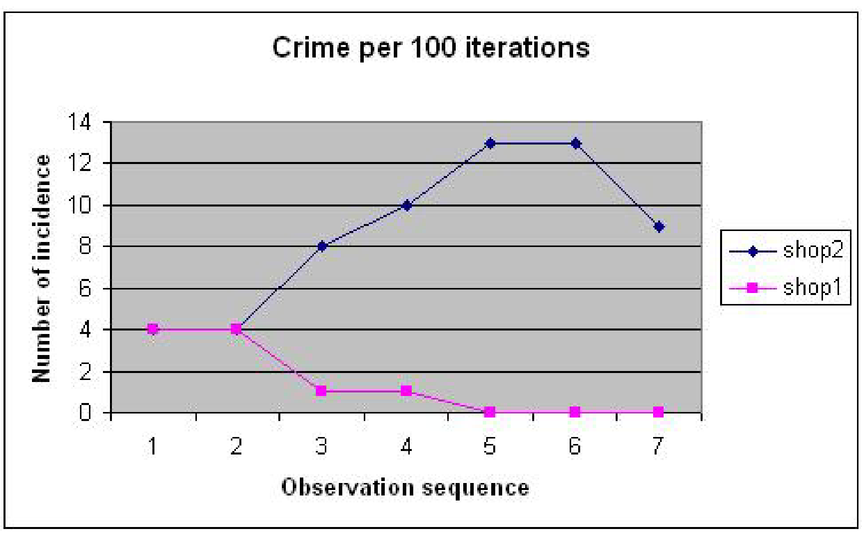


Figure 5.11 Development processes of crime rates at shop1 and shop2 after iteration# 1500

Repeating the simulation for more times

It is worth noting that the outcome in Figure 5.10 and Figure 5.11 is only one possibility. The simulation has been run for 10 times, and for 4 times the simulation turns out that shop 1 is the location with most crime in the whole simulation area. However, the outcome that the two shops have equal or near equal number of crime events has not been observed. This indicates that crime events tend to have a skewed distribution even under the condition of same management effectiveness among crime places, given offenders and targets are adaptable. It also indicates that under the condition of the same management effectiveness, crime patterns are not easy to predict (40% chances for shop 1 to become the hotspot; 60% chances for shop 2 to become hotspot).

Why does the crime distribution observed in our computational laboratory show such a skewed pattern? An intuitive explanation is that crime events have a strong interaction with local situations (mainly local routine activities). Crime events are rare, but each crime event tends to have a strong impact on local situation. For example, when the first crime event occurs at shop 2, it is going to cause many target agents switch to shop 1 and leave shop 2 with insufficient guardianship; also, it is going to draw many other offender agents to shop 2 and increase risks of crime at shop 2. Thus, our simulation in this case shows that, (1) the formation of crime patterns is a self-organized and self-reinforced process; (2) hotspots are a function of offender feedback; and (3) small random disturbances result in very different results. These results observed in the computational laboratory give some interesting insights about what is possibly going on in the real world.

Case 3: The power-function distribution of crime in a set of crime places

This case is set up to test whether crime events distribution in multiple crime places fit a power function. Empirical research found that usually a small proportion of crime places account for a large number of crime events. If we plot the accumulated distribution of crime events in crime places, the curve is in the form of a power function (see Spelman 1995). (A power function is used to describe the non-equilibrium distribution of a phenomenon. A standard power function can be expressed as $Y = a*X^b$. Y refers to the accumulated percentage accounted for by the top X objects.) If our computational laboratory is able to model crime patterns well, then we expect to see the power-function distribution of crime in crime places in experimental scenarios.

The activity nodes and street networks are the same as in scenario two, where we used it to simulate the timing of crime. The activity nodes are 22 shops, which sparsely distributed on the street network. The routine activities of offender agents and target agents are a combination of random movement and shopping, with the transitional probability of random movement to shopping set to 0.01. At shop locations, the management effectiveness is set to be 0.5. At other locations, the management effectiveness is relatively high (5.0).

At first, offenders and targets are all set to be adaptable (offender reward and target cost are both set to 100). The initial value of offender motivation is set to 3.0, and the simulation runs for 100,000 iterations, and the generated crime pattern is save on a grid file and imported into Arcview for analysis. The numbers of crime events at the 22 shops are then recorded on a table, sorted by the rank of the shop according to the number of crime events (for example, the shop with the most crime events is ranked as one). The accumulated percentages of crime events among total crime events are then calculated for each shop (for example, for the shop with rank 3, its accumulated percentage of crime is the summation of number of crime events in shops with rank 1, 2 and 3 then divided by total number of crime events). The accumulated percentages are mapped in Figure 5.12(a).

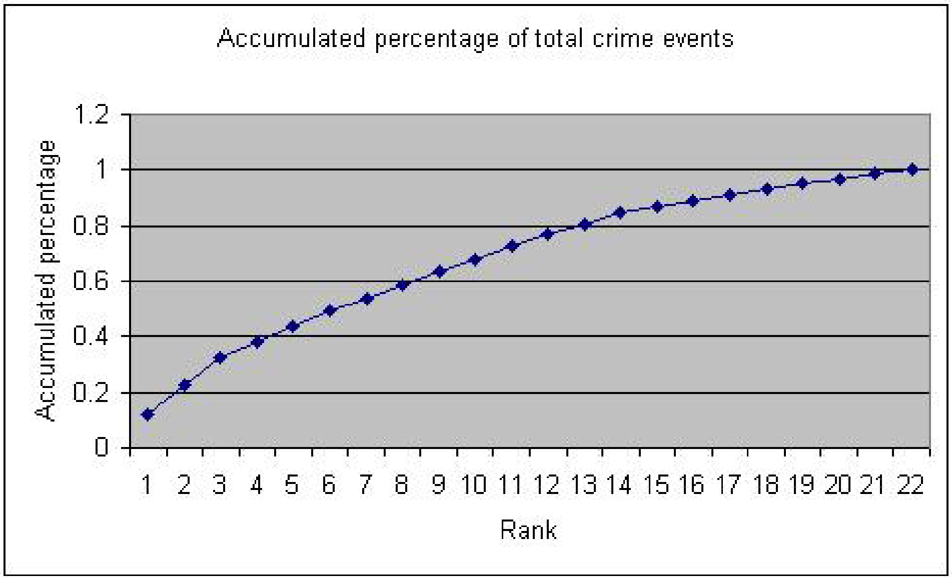


Figure 5.12(a) The accumulated distribution of crime events at the 22 shops when offenders and targets are adaptable

Table 5.6(a) Data set shown in Figure 5.12(a)

Rank (X)	Count of crime	Accumulated percentage (Y)
1	27	0.118
2	25	0.228
3	22	0.325
4	13	0.382
5	13	0.439
6	12	0.491
7	11	0.539
8	11	0.588
9	11	0.636
10	10	0.680
11	10	0.724
12	10	0.768
13	9	0.807
14	9	0.846
15	5	0.868
16	5	0.890
17	5	0.912
18	5	0.934
19	4	0.952
20	4	0.969
21	4	0.987
22	3	1.000

Figure 5.12(a) shows that the distribution of crime events at the 22 shop locations fits a power function. The curved illustrated is similar as the curve made by (Spelman 1995). If the distribution of crime events is even at all shop locations, then we expect the curve to be a straight line. However, the curve in Figure 5.12 is obviously not a straight line. The top 6 crime places (which are only 27% of all crime places) account for about 50% of total crime events. Again, this shows that the distribution of crime events among crime places is highly skewed.

<u>Fit the data against the power function and a contrast study</u>

In order to test whether Figure 12(a) really fits a power function or not, the data set (Table 5.6(a)) of Figure 12(a) is fit against a standard power function $Y = a*X^b$, where Y is accumulated percentage, and X is rank. After a logarithm transformation of both sides of the power function, it becomes logY=loga + b*logX. Thus, we can use a linear regression model to test the fit of the data set against the power function. In order to do so, we need to logarithmically transform both X and Y. The linear regression analysis of the transformed data generates a R^2 of 0.99, with slope equal to 0.66. Note that if b equals to 1.0, then the curve in Figure 12(a) would have been a straight line. The fact that coefficient b is different from 1.0 suggests that the distribution in Figure 5.12(a) is a skewed distribution. The R^2 suggests that the data fits the power function quite well.

Another question is what if offenders and targets are all set to be non-adaptable? Would there still be a skewed distribution of crime events among crime places? In order to make a contrast study, we set all offenders and targets to be non-adaptable (offender reward and target cost are set to be 0). All other parameters are set the same as the former situation (when offenders and targets are all adaptable). Again, the simulation runs for 100,000 iterations. The accumulated distribution of crime events among crime places is shown in Figure 12(b), and the data is presented in Table 5.6(b).

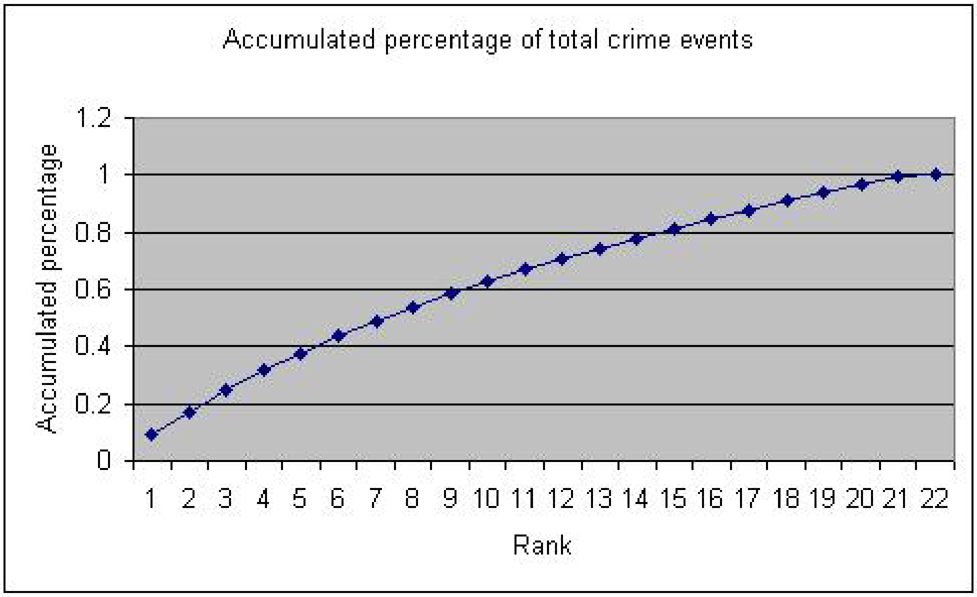


Figure 5.12(b) The distribution of crime events at the 22 shops when offenders and targets are non-adaptable

The data set in Table 5.6(b) is fit against the same power function $Y = a*X^b$ using the same procedures as above. The regression analysis gives an R^2 of 0.99 and a coefficient b of 0.75, which suggests that the distribution is still slightly skewed (b is less than 1.0) and the curve in Figure 5.12(b) still fits a power function quite well.

The regression analysis reveals some interesting difference between the two studies. While the two curves in Figure 5.12(a) and Figure 5.12(b) both fit the power function quite well, their coefficients b are different. When offenders and targets are all adaptable, the coefficient b equals 0.66, which is less than 0.75 (when offenders and targets are non-adaptable). Recall that the closer b is to 1, the straighter the line is. The straighter the line is, the more uniform the distribution is. This suggests that when offenders and targets are all adaptable, the spatial crime pattern tends to be more skewed than when offenders and targets are all non-adaptable. This helps to explain that offender and target adaptation contributes to skewed spatial distribution of crime.

Table 5.6(b) The data set presented in Figure 5.12(b)

Rank	Count of crime	Accumulated percentage
1	109	0.094128
2	90	0.171848
3	90	0.249568
4	77	0.316062
5	70	0.376511
6	69	0.436097
7	60	0.48791
8	59	0.53886
9	57	0.588083
10	48	0.629534
11	45	0.668394
12	44	0.70639
13	43	0.743523
14	40	0.778066
15	39	0.811744
16	38	0.84456
17	37	0.876511
18	37	0.908463
19	36	0.939551
20	34	0.968912
21	31	0.995682
22	5	1

5.4 Scenario four: accessibility and crime

Environmental criminology studies the physical environment and crime. The effect of the urban environment and backcloth on crime has been examined intensively. These studies include examining accessibilities, urban morphology, transportation infrastructures, land uses, environment design, street lighting, and activity nodes and paths. Empirical studies found that street intersections tend to have more crimes than other crime places (see Beavon, Brantingham & Brantingham, 1994; Brantingham & Brantingham 1991).

As a simple scenario of testing the effect of accessibility on crime in our computational laboratory, this experiment tests the hypothesis that street intersections have more crime than other locations. The routine activity nodes used in this scenario are 22 shops on the street network that were used in previous scenarios. Eleven of these shops are on street intersections, and the other eleven shops are within street segments. The layout of the shops is exactly the same as Figure 5.7 of scenario two. Other parameter settings are the same as case 3 of scenario three. The initial value of offender motivation is set to be 3, and the simulation run for 100,000 iterations (about 17 days).

The simulated crime pattern is saved on a grid file and imported to Arcview for analysis. Crime events at street intersections and crime events at other places are recorded in Table 5.7, which becomes a sample of crime data with two categories for statistical analysis.

Table 5.7 Number of crime events at street intersections and block-face locations

Street intersections	6	5	8	9	7	7	9	6	14	8	3
Block-face locations	3	2	3	2	2	2	4	2	4	0	4

One-way ANOVA analysis with the data in Table 5.7 indicates that the two categories are significantly different (P-value = 0.00003). This indicates that street intersections indeed have higher chances of crime than block-face locations. How to explain that street corners tend to have more crime in the simulation experiments? One explanation is that street corners have better accessibility and have more chances for agents to meet with each other and interact with each other that lead to more crime events. This again helps to demonstrate that the computational laboratory is able to generate crime patterns similar as real world observations.

5.5 Scenario five: target adaptability and risks of victimization

A rational assumption of this research is that individuals tend to avoid crime in everyday activities. This is known as "self-policing" behavior (Bottoms, 1994). In our simulation model, target spatial adaptation is realized by giving negative rewards to target victimization templates. According to reinforcement learning theory, negative rewards punish the agent so that the agent tends to avoid risky places in the future. Therefore, target spatial adaptation corresponds to self-policing behavior.

In our simulation modeling, the "self-policing" behaviors are in two senses: one is spatial behavior, such as targets avoid risky places; the other is non-spatial behavior, such as targets change guardian capability and desirability following victimization. Since it is difficult to study the interactions between individuals in the real society (because usually there is no data available in these aspects), it would be meaningful to study such self-policing behavior and its effects on target victimization risk in a simulation environment.

That our computational laboratory supports agent adaptation allows us to experiment with different agent adaptation strategies. The adaptabilities of agents in the computational laboratory have been tested in scenario one. Reasonable adaptation behaviors have been observed. By making use of these adaptable agents, this section is devoted to experiments of target adaptation and its effect on crime rates and victimization rates of the whole society as well as individual target groups.

Case 1: Target spatial adaptation and the effect on victimization rate of the society

The first choice of self-policing is spatial adaptation, which refers to the behavior that targets avoid risky places to avoid victimization. In this case, the victimization rate of the society before target agents are set to be adaptable is compared with the victimization rate after target agents are set to be adaptable to examine whether spatial adaptation helps in reducing crime rates. The street network and activity nodes used in the simulation are the same as in scenario two (the scenario for simulating timing of crime). To focus on target agent adaptability and to make the simulation simple, offender agents can not spatially adapt in this scenario (this is done by set the reward of crime to be 0). All other settings are the same as the settings of scenario two.

The results confirm that target agents' spatial adaptation have an effect on the victimization rate of the society. First, all target agents are set to be non-adaptable, i.e., their cost of crime parameter is set to 0 for all target agents. Under this condition, there are totally about 1200 crime events that occurred for a simulation period of 100,000 iterations (about 17 days). Second, all target agents are set to be adaptable (their cost of crime parameter is set to be 100 for all target agents). The result indicates that there are only about 800 crime events occurred in the same simulation period. A 33% percent reduction of crime rate is resulted after target agents are set to be adaptable.

Case 2: Non-spatial adaptation and the effect on target victimization risks

Another choice of target self-policing is to change states (e.g., guardian capability and desirability in this research). In this case, the effect of target agents changing guardian capability and desirability on target future victimization risks will be examined. All settings are the same as case 1, except that neither offender agents nor target agents can spatially adapt so that we can focus on non-spatial adaptation in this case.

A. Target adaptation of guardian capability

Suppose that target agents tend to increase guardian capability following each victimization incident to enhance self-protection capability (for example, finding a partner to go shopping). In our computational laboratory, this adaptation behavior is modeled by a linear function as follows:

$$\gamma_t = \gamma_0 + kg * V_t \quad \ldots\ldots(5.1)$$

where γ_t is the guardian capability at iteration *t*, and γ_0 is the initial guardian capability. γ_0 is set to 0.5 for all target agents. V_t is the number of victimizations before iteration *t* of the target agent. *kg* is guardian capability coefficient (a float value within range [0, 1]), deciding target agent guardian capability adaptation rate. For example, if *kg* is 0, then target agent will not change guardian capability. Therefore, guardian capability is a positive function of past victimization experience. With more past victimizations, a target agent tends to increase guardian capability to avoid future victimizations.

In order to test the effect of target adaptation of guardian capability on future risk of victimization, the 100 target agents are divided into four groups of 25 agents. The setting of guardian capability coefficients *kg* for the four groups is in Table 5.8. The simulation is then run for 100,000 iterations. Crime events generated in the simulation are summarized in the four groups as shown in Table 5.8 and Figure 5.13.

Table 5.8 Target guardian capability adaptation rates and victimization outcome

Group ID	kg	Victimizations
1	0.00	381
2	0.33	175
3	0.66	135
4	1.00	120

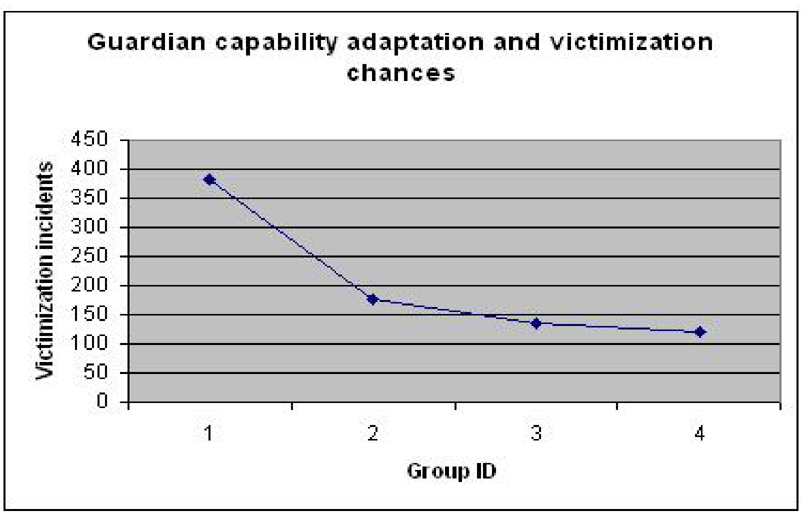


Figure 5.13 Graph representation of Table 5.8

Table 5.8 and Figure 5.13 both show that the more adaptable target agents are, the less the chance of victimization. For example, with guardian capability coefficient set as 0.0, the 25 target agents in group 1 accounts for 381 victimization incidents, while group 4 only accounts for 120 victimization incidents with guardian capability coefficient set as 1.0. This case indicates that guardian capability adaptation is an effective strategy in target self-policing.

B. Target adaptation of desirability

Similar to guardian capability adaptation, targets may change desirability to avoid future victimization, since high desirability tends to attract offender attention. In our computational laboratory, this adaptation behavior is modeled as the following linear function:

$$\delta_t = \delta_0 - kd * V_t \quad \ldots\ldots(5.2)$$

where δ_t is the desirability for a target agent at iteration *t*, and δ_0 is the initial desirability at iteration 0. δ_t is bounded in the range [0, δ_0]. δ_0 is set to be 0.5 for all target agents. *Kd* is desirability coefficient (a float value within range [0, 1]), which determines the target agent's desirability adaptation rate. For example, if *kd* = 0, then the target agent will not change desirability. V_t is agent past victimization experience up to iteration *t*. Therefore, target desirability is a negative function of past victimization experience. With more past victimization incidents, target desirability tends to decrease to avoid future victimization.

Table 5.9 Target desirability adaptation rates and victimization outcome

Group ID	kd	Victimizations
1	0.00	534
2	0.03	96
3	0.06	50
4	0.09	47

To test the effect of target adaptation of desirability on victimization risk, we divide target agents into four groups, each group with 25 target agents. Each group has different *kd* settings, which are listed in Table 5.9. Under this setting, the simulation is run for 100,000 iterations, and the number of victimization incidents for each group is summarized in Table 5.9. The results show that desirability adaptation also affects target victimization risk. For example, group 4 has the highest adaptation rate, which has the lowest victimization incidents.

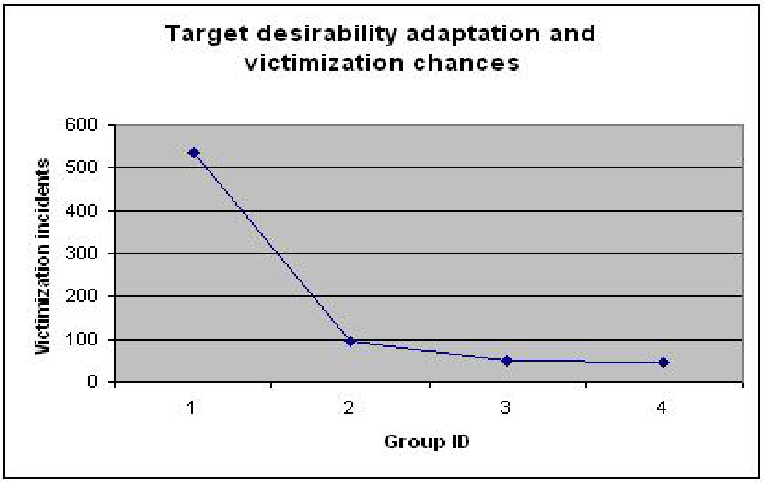


Figure 5.14 Victimization incidents for the four groups of targets with different desirability adaptability

5.6 Scenario six: repeat offending

Empirical studies also found that some offenders commit more crime than others. In fact, just like the repeat location problem, a small proportion of offenders account for a large proportion of crime. This is known as repeat offending problem in environmental criminological literature (Eck, 2001).

Therefore, a natural question asked by criminologists is why some people commit more crimes than other people (Brantingham & Brantingham, 1993)? This section tries to give an explanation on this problem from offender motivation development perspective using our computational laboratory.

Even though motivation is not a central concern in routine activity theory, a motivated offender is a must for any crime event to occur. Offender motivation is a factor used by Eck (1995) in his crime event likelihood evaluation formula. If we regard our computational laboratory as an artificial society, then we can study the development of offender motivation over a "life time" in the simulation. The key is to incorporate some learning theory of offender motivation into the simulation process.

Following the commercial robbery simulation model by Liang, et. al. (2001) and the RA/CA model by Liu, et. al. (2004), our computational laboratory assumes that the learning process of offender motivation follows an "S" curve as defined in equation (5.3) and the graph in Figure 5.14:

$$\mu = \left(\frac{\pi}{2} + \arctan\left(\frac{N_s - N_f}{K_u}\right)\right) / \pi \quad \ldots\ldots(5.3)$$

where, μ is offender motivation value. N_s is the number of successful offense. N_f is the number of failed offenses. K_u is the motivation coefficient, which controls the slope of the 'S' curve. A larger K_u value will make the slope less steep. $\pi/2$ is added to bring the curve up to the x-axis. Finally, the equation is divided by π to make the value range of μ to be [0, 1].

The "S" curve in equation (5.3) and Figure 5.14 mean that when an offender has more experience (either failure or success experience), the change of motivation is less dramatic compared with no experience. Both ends of the "S" curve correspond to more adept offenders. When an offender agent is initialized, the offender agent is at the center of the curve, because the number of successful crimes and the number of failed crimes are both initialized as 0, so $N_s - N_f = 0$. When an offender agent is at the center of the curve, the offender agent is at the stage that learning is fastest.

In order to test whether such a learning function can lead to repeat offending or not, the motivation coefficient is set to be 1.0, and the failure barrier of the crime event is set to be 0.2. All other parameter settings are the same as scenario five. The simulation then runs for 10,000 iterations. The distribution of offenses among offenders is summarized in Table 5.10(a). Figure 5.15 shows the accumulated distribution of offenses among offenders.

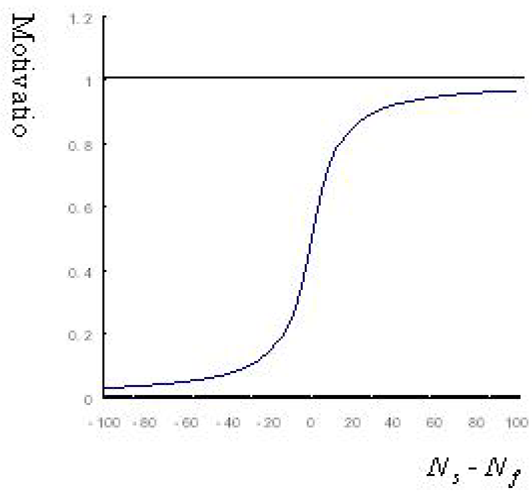


Figure 5.14 Offender motivation learning curve used in the computational laboratory

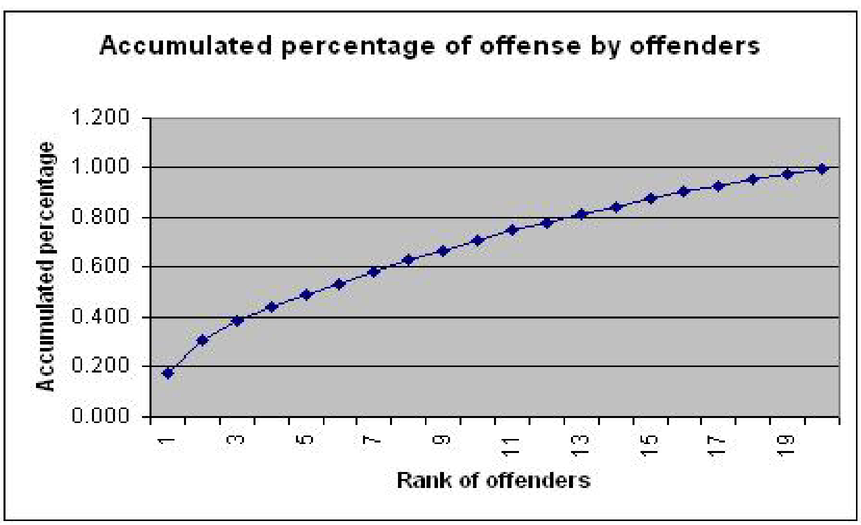


Figure 5.15 The accumulated distribution of offenses among offenders

Table 5.10(a) Accumulated distribution of offenses among the 20 offenders after 10,000 iterations

Rank	ID	Offense	Accumulated percentage
1	10003	22	0.173
2	10017	17	0.307
3	10006	10	0.386
4	10002	7	0.441
5	10010	6	0.488
6	10001	6	0.535
7	10008	6	0.583
8	10013	6	0.630
9	10012	5	0.669
10	10011	5	0.709
11	10004	5	0.748
12	10007	4	0.780
13	10014	4	0.811
14	10005	4	0.843
15	10015	4	0.874
16	10009	4	0.906
17	10019	3	0.929
18	10000	3	0.953
19	10016	3	0.976
20	10018	3	1.000

Figure 5.15 shows that the accumulated distribution of offenses among offenders is similar to a power function. Empirical data analysis by Spelman (1995) reveals a similar curve which is the evidence of repeat offending. From Table 5.10 (a), we know that the top four offenders (20% of offender population) account for about 44% of total offense. Again, the distribution of offenses among offenders is highly skewed.

If we check the offender motivation development process, then it is clear why some offender agents commit more crime than other agents. Offender agent with ID 10017 is one of the top-ranked offenders in terms of number of offenses, and offender agent with ID 10007 is one of the lowest-ranked offenders. Their motivation development is shown in Figure 5.16. It is obvious that starting with the same motivation initial value (both are 0.5), offender 10007 suffered bad luck and decreased motivation value to a low level, while offender 10017 successfully committed a crime and raised motivation value to a high level.

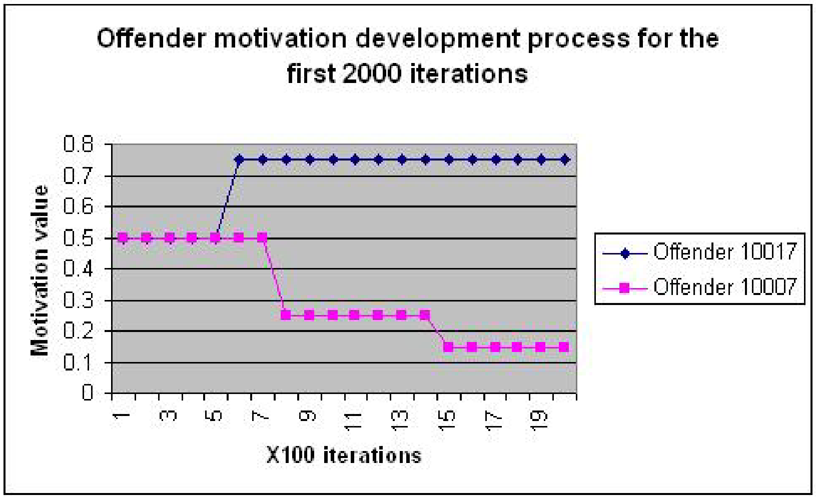


Figure 5.16 Offender motivation development for the first 2000 iterations

Fit the data against the power function and a contrast study

In order to test how well the distribution curve in Figure 5.15 fits a power function, the data in Table 5.10(a) is fit against a standard power function $Y = a*X^b$ using the same procedures as 5.3 case 3. A linear regression analysis with the transformed data comes up with a R^2 of 0.99, and coefficient b equals 0.55. The coefficient b is different from 1.0, which suggests that the distribution is skewed. The R^2 suggests that the curve in Figure 5.15 fits the powerful function quite well.

What if offenders do not have an S-curve learning capability? Would there still be a skewed distribution in offending pattern among offenders? In order to test that the skewed distribution of offenses are indeed caused by the S-curve learning, we need a contrast study that runs the simulation under the condition that offenders do not learn motivation from past experience. In order to disable offender motivation learning, we just need to set the motivation coefficient value to be extremely large, such as 100,000. Thus, the "S" curve in Figure 5.14 would become extremely flat, and offenders are not able to change their motivation. Other parameters remain unchanged. Then the simulation runs for 10,000 iterations. The accumulated distribution of offenses among offenders is shown in Figure 5.17, and the data associated with the distribution is presented in Table 5.10(b).

Table 5.10(b) Accumulated distribution of offenses among offenders under the condition that offenders don't develop motivation

Rank	offenderID	Offense	Accumulated percentage
1	10009	8	0.075
2	10000	8	0.150
3	10001	7	0.215
4	10008	7	0.280
5	10014	6	0.336
6	10013	6	0.393
7	10012	6	0.449
8	10015	6	0.505
9	10003	6	0.561
10	10010	6	0.617
11	10007	5	0.664
12	10018	5	0.710
13	10002	5	0.757
14	10016	5	0.804
15	10004	4	0.841
16	10006	4	0.879
17	10019	4	0.916
18	10005	3	0.944
19	10011	3	0.972
20	10017	3	1.000

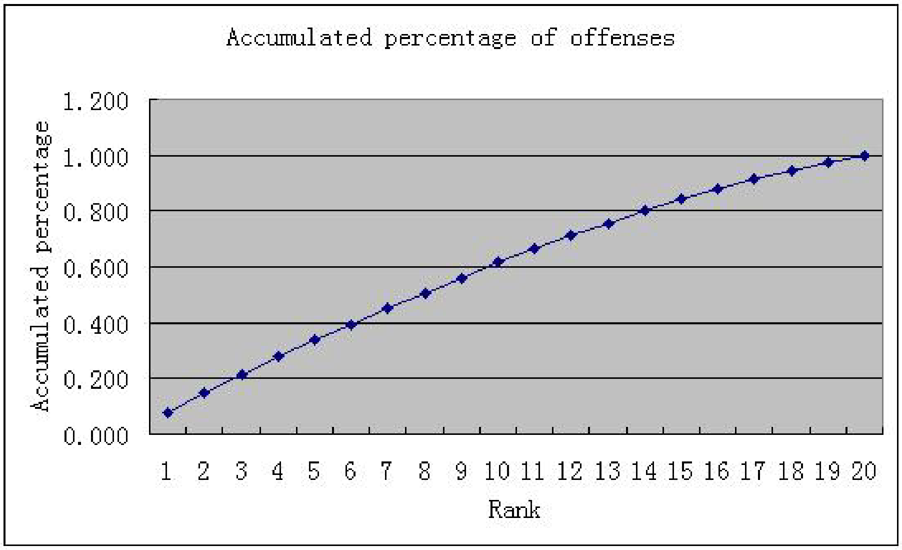


Figure 5.17 Accumulated distribution of offenses among offenders when offenders don't develop motivation

The data in Table 5.10(b) is then fit against the standard power function, which comes up with R2 of 0.99, and coefficient b equals 0.85. The fact that coefficient b is less than 1 shows that the distribution is still skewed. However, compared with the case when offenders learn and develop their motivation, this b value is closer to 1. This means that the distribution of offenses among offenders tends to be more skewed when offenders have learning capabilities with their motivation. This repeat offending pattern simulated in this scenario reveals that the S-curve learning of offender motivation contributes to repeat offending.

5.7 Summary

Among the five scenarios (from scenario two to scenario six), "timing of crime" and "accessibility and crime" are designed to validate the computational laboratory by showing that our computational laboratory is able to generate crime patterns observed in real world. In the "timing of crime" scenario, we observed that crime peaks are associated with the peaks of shopping activities; in the “accessibility and crime” scenario, we observed that street intersections tend to have more crime than block-face locations. All these conform to known crime theories and empirical findings. This shows that the RA/CA/ABM computational laboratory is a valid tool in simulating crime patterns and demonstrating crime theories, which is useful in teaching crime theories.

In the other three scenarios, the computational laboratory is applied to examine the implications of agent adaptations to crime patterns. Scenario three is to examine the effect of agent adaptation on spatial patterns of crime; scenario five examines whether target adaptation affects future victimization risks; scenario six examines the effect of offender S-curve learning on crime patterns associated with offender population. Each of these scenarios provides insights to the mechanisms behind crime patterns. This shows that the RA/CA/ABM computational laboratory can be used in testing crime theories.

Chapter 6 Conclusion

6.1 Summary and conclusion

This research seeks to contribute to crime pattern studies by designing, implementing and testing a computational laboratory to study the interactions among offenders, targets and crime places. Using routine activity theory and crime pattern theory as the theoretical background, the RA/CA/ABM computational laboratory is built using spatial autonomous agents, cellular automata and Object Oriented Programming. The computational laboratory provides a spatial crime simulation model which is derived from routine activity theory. Offender agents and target agents are adaptable in both spatial behaviors and non-spatial properties. The simulation model is then wrapped by a Windows graphic user interface, which allows users to manipulate agent interaction rules and parameters to examine the implications of agent adaptations on crime patterns. Compared with the RA/CA crime simulation model, the contributions of the RA/CA/ABM computational laboratory include using spatial autonomous agents to explicitly model agents' routine activities in space and time, so that agents' spatial behaviors become adaptable. Further, the computational laboratory is an advance over the RA/CA model because a well-designed user interface for the simulation model is now ready, which is independent from any GIS interfaces. The interface provided by the RA/CA/ABM computational laboratory allows users to design spatial crime simulation experiments conveniently.

Examples are given in chapter five to demonstrate the usefulness of the RA/CA/ABM computational laboratory. These simulation experiments successfully generated some crime patterns that are known in real world, such as the timing of crime and that street intersections have more crime than block-face locations. Further, the effect of offender agents and target agents adaptations to crime patterns is examined. Power function distributions of crime in crime places and in offender population have been observed and analyzed.

Major conclusions of this dissertation research include the following points:

1. A reinforcement learning algorithm can be used to design spatial autonomous agents with the capability of wayfinding and navigation.

2. A loose coupling of agent-based models with GIS is an efficient way of developing space-time dynamic simulation models, which combines the advantage of agent-based modeling in dynamic spatial modeling and the advantage of GIS in spatial representation.

3. Agent-based modeling is a useful and promising approach for environmental criminological research and crime pattern simulation.

4. The RA/CA/ABM computational laboratory developed in this research produces spatial, temporal, offending, and victimization patterns that mimic real world patterns, thus showing it is a valid crime pattern simulator.

5. Crime hotspot development is a function of poor management. In the simulation scenario with two crime places and non-equivalent management effectiveness, the place with a poorer management effectiveness had an 80% chance of becoming a hotspot.

6. Crime hotspot development is a function of offender and target adaptation. In the scenario with two crime places and offenders and targets are both set to be adaptable, a crime hotspot shows up despite the management effectiveness being equal between the two places.

7. Offender agents and target agents' adaptations contribute to the power-function distribution of crime events in crime places. Also, offender agents' motivation adaptation contributes to the power-function distribution of crime events among the offender population.

8. Targets' adaptation is a useful strategy in reducing target victimization risks.

6.2 Contributions

Contributions to the SPatial Adaptive Crime Event Simulation (SPACES) project

SPatial Adaptive Crime Event Simulation (SPACES) is a collaborative project between the Department of Geography and Criminal Justice Division at the University of Cincinnati. The long-term goal of the SPACES project is to develop a spatial crime event and crime pattern simulation model that can be used for: (1) theoretical experimentation with crime event theories and hypotheses; (2) generate new hypotheses for empirical studies; (3) teach crime theories in classroom; (4) bench test crime prevention policies and strategies prior to implementation. The position of SPACES within three disciplines is illustrated in Figure 6.1.

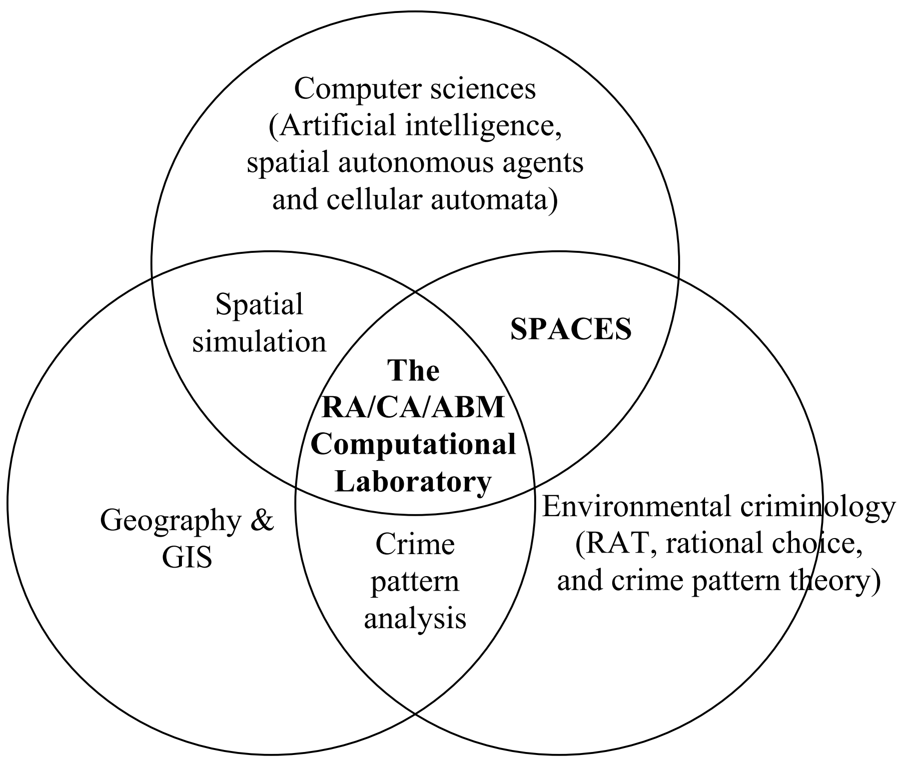


Figure 6.1 The position of SPACES and the RA/CA/ABM computational laboratory within three disciplines

The computational laboratory for crime event and crime pattern simulation introduced in this dissertation research represents one of the first implementations of SPACES ideas. While the full goal of SPACES is still far from being achieved, the first three goals have been achieved in this research. We are able to use the RA/CA/ABM computational laboratory in testing crime theories and hypotheses, generating new hypotheses for empirical studies, and assist in teaching crime theories. The relationship between the RA/CA/ABM computational laboratory and SPACES is illustrated in Figure 6.1.

Contributions to experimental geography and environmental criminology

Computational laboratories provide experimental platforms for geographic research in a number of areas, such as exploring the diffusion of a phenomenon in spatial networks (Dibble, 2001), the growth of urban area (Batty and Xie, 1999), and dynamic distribution of hikers in recreation area (Gimblett, 2002). This dissertation research develops a computational laboratory for spatial crime pattern simulation, which is a new application area for computational laboratories.

The RA/CA/ABM computational laboratory contributes to routine activity theory. The static routine activity theory focuses on conditions that lead to crime event. The RA/CA/ABM computational laboratory makes routine activity theory dynamic by incorporating offender and target adaptations both in spatial behavior and non-spatial properties, so that we can test the effect of agent adaptations on crime patterns. The computational laboratory also puts routine activity theory into a spatial simulation environment, thereby providing a link between crime event theories and crime place theories.

Contributions to GIS and spatial simulation modeling

GIS has been conventionally configured as a spatial database management system (Wu, 1999). GIS is relatively weak in space-time dynamic simulation modeling. The RA/CA/ABM computational laboratory developed in this dissertation research can be used as an extension of commercial GIS software (such as ESRI Arcview GIS) for crime pattern simulation purposes.

This research developed a type of spatial autonomous agent based on reinforcement learning, and applied such spatial autonomous agents in simulation studies to human geography areas. To the best of my knowledge, this type of spatial autonomous agents is new to human geography modeling.

Technically, this research suggests a new algorithm for finding a least cost surface using a multi-agent based approach. The least cost surface has been used as an agent cognitive map, which is learned and improved during the simulation. This indicates that a mulit-agent based distributed computational approach can be used in finding the least cost surface that is usually computed using regular graph theory. However, the formal algorithm has not been developed in the current research and future work is required on this topic.

6.3 Limitations and future work

The RA/CA/ABM computational laboratory presented in this dissertation is only a prototype model for theoretical crime pattern simulation purposes. Much work remains to be done in order to improve it.

First, one of the goals of the SPACES project is to develop a simulation model that can be used to simulate real crime patterns and to bench test crime prevention policies before implementation. Current RA/CA/ABM computational laboratory has only been tried in simulating theoretical crime patterns. The simulation experiments show that agent-based modeling is a plausible approach in crime pattern simulation. It is promising to use similar agents to simulate real crime patterns in an urban area. In order to do so, we need detailed routine activities of the area, and information about the physical environment (including transportation infrastructure, activity nodes, etc.). In addition, more realistic crime event rules needs to be identified for agents.

Second, current implementation of the RA/CA/ABM computational laboratory is insufficient in the modeling of police agents. The model design has taken police agents into consideration, but police agents are treated as ordinary agents like offenders and targets. We did not include any experiments with police agents in chapter five. How to scheduling their patrolling routine into the simulation model needs to be researched in the future.

Third, a systematic exploration of the parameter space needs to be addressed in the future. The effect of different parameter settings on crime patterns needs to be examined and explained. This exploration process will result in more interesting scenarios, and enhance our understanding with the model.

Fourth, the crime simulation scenarios addressed in chapter five provide some insights into crime patterns. However, there are many interesting patterns that have not been simulated. For example, the simulation experiments show that the distribution of crime events among crime places tends to be skewed and stable. However, this result is under the condition of no external intervention (such as police reaction). What would happen to the crime patterns after the introduction of external intervention has not been explored. For example, if police react to crime events by changing policing areas, what will happen to the crime patterns? Will the crime patterns become unstable and non-equilibrium? Such interesting scenarios need to be simulated in the future with our computational laboratory.

Finally, since the RA/CA/ABM computational laboratory is new to the academic community of environmental criminology, it needs future examination and testing to validate it and explore its utility.

List of References

Anselin, L., J. Cohen, D. Cook, W. Gorr, and G. Tita (2000). "Spatial Analysis of Crime". In David Duffee (ed.), *Criminal Justice 2000: Volume 4. Measurement and Analysis of Crime and Justice*. Washington, DC: National Institute of Justice. Pp. 213–262

Barnes, G. C. (1995). "Defining and Optimizing Displacement". In *Crime and Place: Crime Prevention Studies Volume 4*, edited by J. E. Eck and D. Weisburd. Monsey: Criminal Justice Press

Batty, M., Y. Xie, and Z. Sun (1999). "Modeling Urban Dynamics through GIS-based Cellular Automata." *Computers, Environment and Urban Systems* 23, 205-233

Block, R. and C. R. Block (2000). "The Bronx and Chicago: Street Robbery in the Environs of Rapid Transit Stations." In *Analyzing Crime Patterns: Frontiers of Practice*, edited by V. Goldsmith, P. G. McGuire, J. H. Mollenkopf, and T.A., pp. 137-152. Ross. Thousand Oaks: Sage

Blue, V. J. and J. L. Adler (2001). "Cellular automata microsimulation for modeling bi-directional pedestrian walkways." *Transportation Research Part B: Methodological* 35: 293-312

Bonabeau, E. (2002). "Agent-based modeling: methods and techniques for simulating human systems". Proceedings of the National Academy of Science of the United States of America, May 14, 2002. Vol 99. http://www.pnas.org/cgi/reprint/99/suppl_3/7280.pdf

Bottoms, A. E. (1994). "Environmental Criminology". In M. Maguire, R. Morgan, and R. Reiner (Eds.), *The Oxford handbook of criminology*. Oxford: Clarendon Press; New York: Oxford University Press

Brantingham, P. L. and P. J. Brantingham (2003). "Computer Simulation as a Tool for Enviornmental Criminologists". Paper presented on 11th International Symposium on Environmental Criminology and Crime Analysis

Brantingham, P. L. and P. J. Brantingham (1993). "Environment, routine, and situation: Toward a pattern theory of crime". In R. V. Clarke and M. Felson (eds.), *Routine Activity and Rational Choice: Advances in Criminological Theory, Vol. 5.* New Brunswick, NJ: Transaction Publications

Brantingham, P. J. and P. L. Brantingham (1991). "Notes on the Geometry of Crime". In P. J. Brantingham & P. L. Brantingham (Eds.), *Environmental Criminology*. Waveland Press, Inc.: Prospect Heights, Illinois

Brantingham, P. J. and P. L. Brantingham (1991). "Notes on Environmental Criminology". In P. J. Brantingham & P. L. Brantingham (Eds.), *Environmental Criminology*. Waveland Press, Inc.: Prospect Heights, Illinois

Brantingham, P. J. and P. L. Brantingham (1991). "Introduction: The Dimensions of Crime". In P. J. Brantingham & P. L. Brantingham (Eds.), *Environmental Criminology*. Waveland Press, Inc.: Prospect Heights, Illinois

Brantingham, P. J. and P. Brantingham (2001). "The implications of the criminal event model for crime prevention". In *The Process and Structure of Crime: Criminal Event and Crime Analysis, Advances in Criminological Theory, Volume 9*. Edited by R. F. Meier, L. W. Kennedy, and V. F. Sacco. Transaction Publishers: New Brunswick, New Jersey

Brantingham, P. J. and P. Brantingham (1984). *Patterns in Crime*. New York: Macmillan; London: Collier Macmillan

Burstedde, C., K. Klauck, A. Schadschneider, and J. Zittarz (2001). "Simulation of pedestrian dynamics using a two-dimensional cellular automaton". *Physica A 295*, pp. 507-525

Casti, J. L. (1999). "The Computer as a Laboratory". *Complexity*, Vol. 4, No. 5, pp. 12-14. http://journals.ohiolink.edu/local-cgi/send-pdf/031121175035510554.pdf

Caswell, H. and John, M. (1992). "From the Individual to the Population in Demographic Models". *Individual-Based Models and Approaches in Ecology*. DeAngelis and Gross: 36-61

Clarke, K. C. and S. Hoppen (1997). "A Self-modifying Cellular Automaton Model of Historical Urbanization in the San Francisco Bay Area." *Environment and Planning B: Planning and Design* 24, 247-261

Clarke, K. C. and L. J. Gaydos (1998). "Loose-coupling A Cellular Automata Model and GIS: Long-term Urban Growth Prediction for San Francisco and Washington/Baltimore." *International Journal of Geographical Information Science* 12, 699-714

Clarke, K. C., P. Riggan, and J. A. Brass (1995). "A cellular automaton model of wildfire propagation and extinction." *Photogrammetric Engineering and Remote Sensing* 60: 1355-1367

Clarke, R. V. (1992). "Part I: Introduction". In *Situational Crime Prevention: Successful Case Studies*, Edited by R. V. Clarke, pp. 3-36. New York: Harrow and Heston

Clarke, R. V. and M. Felson (1993). "Introduction: Criminology, Routine Activity and Rational Choice". In R. V. Clarke and M. Felson (eds.), *Routine Activity and Rational Choice: Advances in Criminological Theory, Vol. 5.* New Brunswick, NJ: Transaction Publications

Clarke, R. V. (1994). "Situational Crime Prevention: Its Theoretical Basis and Practical Scope". In *Crime Displacement: the Other Side of Prevention*. Edited by Robert P. McNamara, Cummings & Hathway Publishers: NY

Cohen, L. E., and M. Felson (1979). "Social Change and Crime Rate Trends: A Routine Activity Approach." *American Sociology Review* 44, 588-608

Cordella, P., and L. Siegel (Eds.) (1996). *Readings in contemporary criminological theory*. Boston: Northeastern University Press

Cornish, Derek B. and Clarke, Ronald, V. (1986). "Situational Prevention, Displacement of Crime and Rational Choice Theory". in Kevin Heal and Gloria Laycock (eds.), *Situational Crime Prevention: From Theory into Practice*, Her Majesty's Stationery Office, London

Cornish, Derek B. and Clarke, Ronald V. (1987). "Understanding Crime Displacement: An Application of Rational Choice Theory". *Criminology*, Vol. 25, No. 4, pp. 933-47

Cornish, D. B. and R. V. Clarke (1987). "Understanding crime displacement: an application of rational choice theory". *Criminology*. Vol. 25, No. 4

Couclelis, H. (1997). "From cellular automata to urban models: new principles for model development and implementation". *Environment and Planning B*: vol 24, pp165-174
Dibble, C. (2001). "Theory in a Complex World: GeoGraph Computational Laboratories". Ph.D. dissertation, Geography Department at the University of California, Santa Barbara

Doran, J. (2000). "Questions in the Methodology of Artificial Societies". *Tools and Techniques for Social Science Simulation*. Physica-Verlag, Heidelberg: 17-25

Dibble, C. (2001). "Theory in a Complex World: Geograph Computational Laboratories". Ph.D. dissertation, Department of Geography, University of California at Santa Barbara

Eck, J. E. (1995). "Examining Routine Activity Theory: A Review of Two Books." *Justice Quarterly* 12, 783-797

Eck, J. E. and D. Weisburd (1995). "Crime Places in Crime Theory." In *Crime and Place: Crime Prevention Studies Volume 4*, edited by J. E. Eck and D. Weisburd, pp. 1-33. Monsey: Criminal Justice Press

Eck, J. E., J. S. Gersh, and C. Taylor (2000). "Finding Crime Hotspots Through Repeat Address Mappping." In *Analyzing Crime Patterns: Frontiers of Practice*, edited by V. Goldsmith, P. G. McGuire, J. H. Mollenkopf, and T.A., pp. 49-62. Ross. Thousand Oaks: Sage

Eck, J. E. (1994). "The Threat of Crime Displacement". In *Crime Displacement: the Other Side of Prevention*. Edited by Robert P. McNamara, Cummings & Hathway Publishers: NY

Eck, J. E. (1995). "A general model of the geography of illicit retail marketplaces". In J. Eck & D. Weisburd (Eds.), *Crime and place: Crime prevention studies Vol. 4.*. Monsey, NY: Criminal Justice Press

Eck, J. E. (1997). "What do those dots mean? Mapping theories with data?" In T. McEwen & D. Weisburd (Eds.), *Crime mapping and crime prevention*. Monsey, NY: Criminal Justice Press

Eck, J. E. (2001). “Policing and crime event concentration”. In *The Process and Structure of Crime: Criminal Event and Crime Analysis, Advances in Criminological Theory, Volume 9*. Edited by R. F. Meier, L. W. Kennedy, and V. F. Sacco. Transaction Publishers: New Brunswick, New Jersey

Eck, J. E. (2003). “Police Problems: The Complexity of Problem Theory, Research and Evaluation”. In *Problem-Oriented Policing: From Innovation to Mainstream*. Crime Prevention Studies, vol. 15, edited by J. Knutsson. Monsey, New York: Criminal Justice Press

Edmonds, B. (2000). “The Use of Models – Making MABS More Informative”. *Multi-Agent-Based Simulation, Second International Workshop, MABS 2000*. Springer-Verlag, New York: 15-32

Epstein., J. M. and R. Axtell (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. MIT Press: Cambridge, M.A

Epstein., J. M (1999). “Agent-based simulation models and generative social science”. *Complexity*, Vol4, No.5

Fang, W., L. Yang, and W. Fan (2003). “Simulation of bi-directional pedestrian movement using a cellular automata model”. *Physica* A 321, pp. 633-640

Felson, M. (1998). *Crime and Everyday Life*, Second Ed. Pine Forge Press: Thousand Oaks

Felson, M. (1987). “Routine Activities and Crime Prevention in the Developing Metropolis”. *Criminology*. Vol. 25, No. 4

Felson, M. (1995). “Those who discourage crime”. *Crime and place: Crime prevention studies Vol. 4.* Monsey, NY: Criminal Justice Press

Ferrand, N. (1996). “Modeling and Supporting Multi-Actor Spatial Planning Using Multi-Agents Systems”. 3d National Center for Geographic Information end Analysis (NCGIA) Conference on "GIS and Environmental Modelling", SantaFe, 1996, Santa Barbara, CA. http://wwwlisc.clermont.cemagref.fr/Labo/MembresEtPagesIntermediaires/pagesperso/membres_actuels/ferrand/santefe.htm

Gabor, T. (1994). “Crime Displacement and Situational Prevention: Toward the Development of Some Principles”. In *Crime Displacement: the Other Side of Prevention*. Edited by Robert P. McNamara, Cummings & Hathway Publishers: NY

Gilbert, N. (2000). “Models, Processes and Algorithms: Towards A Simulation Toolkit”. *Tools and Techniques for Social Science Simulation*. Physica-Verlag, Heidelberg: 3-16

Gimblett, H. R. (2002). “Integrating geographic information systems and agent-based technologies for modeling and simulating social and ecological phenomenon”. *Integrating*

Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes. Oxford University Press: New York

Golledge, R.G. and R. J. Stimson (1997). *Spatial behavior: a geographic perspective*. NY: The Guilford Press

Gunderson, L. and D. Brown (2000). "Using a Multi-Agent Model to Predict Both Physical and Cyber Criminal Activity". http://vijis.sys.virginia.edu/publication/SMCMultiAgent.pdf

Hagerstrand, T. (1967). *Innovation diffusion as a spatial process*. Chicago: University of Chicago Press

Hollway, W. and T. Jefferson (2000). "The Role of Anxiety in Fear of Crime." In *Crime Risk and Insecurity,* edited by T. Hope and R. Sparks, pp. 31-49. London and New York: Routledge

Itami, R. M. (2002). "Mobile agents with spatial intelligence". *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes.* Oxford University Press: New York

Jiang, B. and H. R. Gimblett (2002). "An agent-based approach to environmental and urban systems within geographic information systems". *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes.* Oxford University Press: New York

Jeffery, C. R. and Zahm, D. L. (1993). "Crime prevention through environmental design, opportunity theory, and rational choice models". In R. V. Clarke and M. Felson (eds.), *Routine Activity and Rational Choice: Advances in Criminological Theory, Vol. 5.* New Brunswick, NJ: Transaction Publications

Johnston, R., (1997), "Behavioural Geography", in *Geography and Geographers: Anglo-American Human Geography since 1945*, London: Arnold, 147-172
Kennedy, L. W. and S. W. Baron (1993). "Routine activities and a subculture of violence: a study of violence on the street". *Journal of Research in Crime and Delinquency*, 30, pp. 88-112

Kennedy, L. W., and E. G. V. Brunschot (2001). "Routines and the Criminal Event". In *The Process and Structure of Crime: Criminal Event and Crime Analysis, Advances in Criminological Theory, Volume 9*. Edited by R. F. Meier, L. W. Kennedy, and V. F. Sacco. Transaction Publishers: New Brunswick, New Jersey

Kozma, R. and P. Ankaraju (2003). "Learning spatial navigation using chaotic neural network model". International Joint Conference on Neural Networks IJCNN'2003, Portland, OR, July 14-19, 2003. http://cnd.memphis.edu/sodas/publications/2003/conference/ijcnn03-kozma.pdf

Liang, J., L. Liu and J. Eck. (2001). "Simulating Crimes and Crime Patterns Using Cellular Automata and GIS," *UCGIS Student Award Paper 2001.* http://www.ucgis.org/f2oppor.html

Liu, L., X. Wang, J. Eck and J. Liang (2004). “Simulating Crime Events and Crime Patterns in a RA/CA Model”. *Geographic Information Systems and Crime Analysis*, edited by F. Wang, Idea Group Inc.: Hershey, PA

Lawrence, P. Macario Mendoza, S. M. Manson, Y. Ogneva-Himmelberger, et al. (2001). “Deforestation in the southern Yucatan peninsular region: An integrative approach”. *ForestEcology and Management* 154(3): 353-370

Lurigio, A. J. and D. P. Rosenbaum (1994). “Evaluation Research in Community Crime Prevention: A Critical Look At the Field”. In *Crime Displacement: the Other Side of Prevention*. Edited by Robert P. McNamara, Cummings & Hathway Publishers: NY

Manson, S. M. (2000). “Agent-based dynamic spatial simulation of land-use/cover change in the Yucatan Peninsula, Mexico”. Paper presented on 4th international conference on integrating GIS and Environment Modeling (GIS/EM4): Problems, prospects and research needs. http://140.232.1.5/~smanson/Manson_2000_GISEM4_ADSS_www.pdf

Meier, R. F., L. W. Kennedy, and V. F. Sacco (2001). “Crime and the criminal event perspective”. In *The Process and Structure of Crime: Criminal Event and Crime Analysis, Advances in Criminological Theory, Volume 9*. Edited by R. F. Meier, L. W. Kennedy, and V. F. Sacco. Transaction Publishers: New Brunswick, New Jersey

Meier, R. F. (2001). “Macro Models of Criminal Events”. In *The Process and Structure of Crime: Criminal Event and Crime Analysis, Advances in Criminological Theory, Volume 9*. Edited by R. F. Meier, L. W. Kennedy, and V. F. Sacco. Transaction Publishers: New Brunswick, New Jersey

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill: New York

Morrill, R., G. L. Gaile, and G. I. Thrall (1988). Spatial diffusion. Newbury Park, CA: Sage

Muller, G. (2000). “Computer Assisted Interfacing: On the Use of Computer Simulation for Theory Construction”. *Tools and Techniques for Social Science Simulation*. Physica-Verlag, Heidelberg: 26-47

Murray, A. T. and V. Estivill-Castro (1998). “Cluster discovery techniques for exploratory spatial data analysis". *International Journal of Geographical Information Science* 12, 431-443

Murray, A. T. , I. McGuffog, J.S. Western and P. Mullins (2001). “Exploratory spatial data analysis techniques for examining urban crime”. *British Journal of Criminology* , 41, 309-329

Norling, E., Sonenberg, L. and Ronnquist, R. (2000). “Enhancing Multi-Agent Based Simulation with Human-Like Decision Making Strategies”. *Multi-Agent-Based Simulation, Second International Workshop, MABS 2000*. Springer-Verlag, New York: 214-228

Norris, F. H. (1997). “The Psychological Consequence of Crime.” In *Victims of Crime*, edited by R.C. Davis, A. J. Lurigio and W. G. Skogan, pp. 146-166. Thousands Oaks, London: Sage

Olligschlaeger, A.M. (1997). “Spatial Analysis of Crime Using GIS-Based Data: Weighted Spatial Adaptive Filtering and Chaotic Cellular Forecasting with Applications to Street Level Drug Markets”. Doctoral dissertation of Carnegie Mellon University

Openshaw, S. and C. Openshaw (1997). Artificial intelligence in geography. Chichester; New York: Wiley

Parker, D. C., S. M. Manson, M. A. Janssen, M. J. Hoffmann, and P. Deadman (2003). “Multi-agent systems for the simulation of land-use and land-cover change: a review”. Annals of the Association of American Geographers, Vol. 93, pp. 314-337

Paternoster, R. and S. Simpson (1993). “A Rational Choice Theory of Corporate Crime”. In R. V. Clarke and M. Felson (eds.), *Routine Activity and Rational Choice: Advances in Criminological Theory, Vol. 5.* New Brunswick, NJ: Transaction Publications

Peuquet, D. J. (2002). *Representations of space and time*. New York: the Guilford Press

Phipps, M. and A. Langlois (1997). “Spatial dynamics, cellular automata, and parallel processing computers”. *Environment and Planning B*: vol 24. pp193-204

Ratcliffe, J. H. and M. J. McCullagh (1999). “Hotbeds of crime and the search for spatial accuracy”. *Journal of Geographical Systems* (1999) 1: 385-398

Rauch, J. (2002). “The World on a Screen”. *The Atlantic*. http://www.theatlantic.com/unbound/interviews/int2002-03-29.htm

Rauch, J. (2002). “Seeing Around Corners”. *The Atlantic*. Vol. 289, No. 4, pp. 35-48

Rodrigues, A. (1997). “Defining spatial agents”. In Spatial Multimedia and Virtual Reality Research Monograph (to be published by Taylor and Francis). http://gasa.dcea.fct.unl.pt/gasa/gasa98/gasa98/phd/armanda/liv98.html

Rodrigues, A. C. Grueau, J. Raper, and N. Neves (1998). “Environmental Planning using Spatial Agents”. Innovations in GIS 5, Taylor & Francis, to appear. http://gasa.dcea.fct.unl.pt/gasa/gasa98/gasa98/phd/ armanda/rodrigues.html

Schelhorn, T., D. O'Sullivan, M. Haklay and M. Thurstain-Goodwin (1999). “STREETS: an agent-based pedestrian model”. Paper presented at *Computers in Urban Planning and Urban Management*, Venice, 8–10 September. [Also available as *CASA Working Paper 9* at http://www.casa.ucl.ac.uk/streets.pdf

Sherman, L., Gartin, P., & Buerger, M. (1989). “Hotspots of predatory crime: Routine activities and the criminology of place”. *Criminology*, 27, 27-55

Stafford, H. A. (2000). “Crime Mapping: Gaps between Theory and Method”. Presentation at the Fourth Annual International Crime Mapping Research Conference: *Wheredunit? Investigating the Role of*

Place in Crime and Criminality. Website:
http://www.ojp.usdoj.gov/nij/maps/Conferences/00conf/Presentations/Stafford.ppt

Sutton, R. S. and A. G. Barto (1998). *Reinforcement Learning: An Introduction*. Cambridge, Mass.: MIT Press

SWARM development group (2004). Documentation Set for SWARM 2.1.1. http://www.swarm.org/swarmdocs/set/set.html

Swartz, C. (2000). "The Spatial Analysis of Crime: What Social Scientists Have Learned." In *Analyzing Crime Patterns: Frontiers of Practice*, edited by V. Goldsmith, P. G. McGuire, J. H. Mollenkopf, and T.A., pp. 33-46. Ross. Thousand Oaks: Sage

Takeyama, M., and H. Couclelis (1997). "Map dynamics: integrating cellular automata and GIS through Geo-Algebra", *International Journal of Geographical Information Science*, 11, 1: 73-91

Taylor, R. A. J. (1992). "Simulating populations obeying Taylor's Power Law". *Individual-Based Models and Approaches in Ecology*. DeAngelis and Gross: 295-311

Tomlin, D. (1990). *Geographic Information Systems and Cartographic Modeling*. Prentice-Hall: Englewood Cliffs, New Jersey

Turner II, B. L., S. Cortina Villar, D. Foster, J. Geoghegan, E. Keys, P. Klepeis, D.

Ward, R. H., T. J. Ward, and J. Feeley (1975). "Chapter 1. Introduction". In *Police Robbery Control Manual*. U.S. Department of Justice

Weiss, G. (1999). *Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence*. MIT Press: Cambridge, Mass.

Westervelt, J. D. (2002). "Geographic information system and agent-based modeling". *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes.* Oxford University Press: New York

Wilson, J. Q. (1983). "Thinking about crime: the debate over deterrence". *The Atlantic*. Vol. 252, No. 3; pp. 72-88

Wilson, J. Q. and G. L. Kelling (1982). "Broken Windows: the police and neighborhood safety". *The Atlantic*. Vol 249, No.3; pp. 29-38

White, N. M. (2003). *Agent-based simulation for predictive modeling: a study of economically motivated crime in Charlotte, NC*. Master thesis. Department of computer science, the University of Virginia

Wright, R. T. and S. H. Decker (1997). *Armed Robbers in Action: Stickups and Street Culture*. Northeastern University Press

Wu, F. (1999). "GIS-based Simulation as An Exploratory Analysis for Space-time Processes." *Journal of Geographical Systems* 1: 199-218

Appendix A:

User guide to the program of RA/CA/ABM computational laboratory

I. Introduction

The RA/CA/ABM computational laboratory is designed for studying the space-time interaction between offenders, targets, crime controllers, and crime places; exploring the implications of crime theories on crime patterns at micro crime place level, and on crime patterns among offender and target populations; and assisting teaching routine activity theory and crime pattern theory.

Each simulation will create a number of offender agents, target agents, and police agents. These agents are then released on the street network to execute user-defined routine activity schedules. Agent routine activities involves a street network and different types of routine activity nodes. Street network and routine activity nodes are specified by users before the simulation starts. Offender agents, target agents, and police agents then interact on the street network. Crime events resulting from the simulation will be put on a log file, and the accumulated spatial pattern will be saved in a grid file.

Using the software needs some basic GIS skills, such as editing data, converting data types, and importing/exporting grid data.

This user guide will provide information on installation, data requirement and data preparation, user interface elements, creating new simulation, setting activity schedules, setting simulation parameters, opening/saving parameters settings, simulation control, output format and naming, and runtime querying agent properties and crime patterns. After reading this user guide, users should be able to use this program quite well.

II. Installation

Installation from CD:

1. Directly copy the folder "GridWorldWin" to the root directly of any local drive. For example, C:\ or D:\.
2. Executable file name is "GridWorldWin.exe". Double click it to run the program.
3. Street network, activity nodes and activity schedules should be put into the "Data" folder under "GridWorldWin" folder. There are some data come with the CD which is already in the "Data" folder.
4. There should be at least 10 megabytes free space on local drive, depending on the size of the simulation area.

III. Creating a new simulation

This section demonstrates the procedures of creating a new simulation using the simulation program through an example scenario, and describes the interface elements of the simulation program (including control buttons, indicators, dialogues, and windows of the simulation program). Creating a new simulation sometimes is also referred to experiment design using the simulation program.

<u>1. Goal of the example</u>

The goal of this example is to create a simulation to demonstrate offender and target agent adaptability in routine activities. Target agents are set to run from a bus station to a shop, to work place and then return to the bus station. Offender agents are set to “hang around” the shops on the street network. Offenders can attack targets when they converge in space and time. This process will loop until the specified number of iterations are finished. We will see how offender agents and target agents adapt according to the environment change.

2. Data requirements

Three types of data needs to be prepared before starting the program, which are the street network grid, routine activity nodes, and routine activities schedule. Street network is a grid file converted from vector line file. Routine activity nodes are also converted into grid data from point vector data. These grid data should be in a binary raster format (with extension name “.flt”) that can be read by the program and imported into Arcview. Routine activity schedules are matrixes of transitional probabilities save in a directory. For details of preparing these data, see later sections. These data should be put on local drive, usually in the “Data” subfolder of “GridWorldWin”.

The path and file name of these data will be put into the simulation configuration dialogue, and these data will finally be read by the simulation program to create a simulation. How to put the path and name of these data into the simulation configuration dialogue will be discussed later.

3. Starting the program

First, we start the simulation program by double-clicking “GridWorldWin.exe”, and the program mainframe shows up (Figure A1). At this point, most of the buttons gray out. The only buttons that can be used now are those buttons labeled in Figure A1.

The functions of button 1, 2 and 3 are as follows:

(a) Button 1 – creating new simulation;

(b) Button 2 – opening existing simulation;

(c) Button 3 – displaying program information and copyright statement.

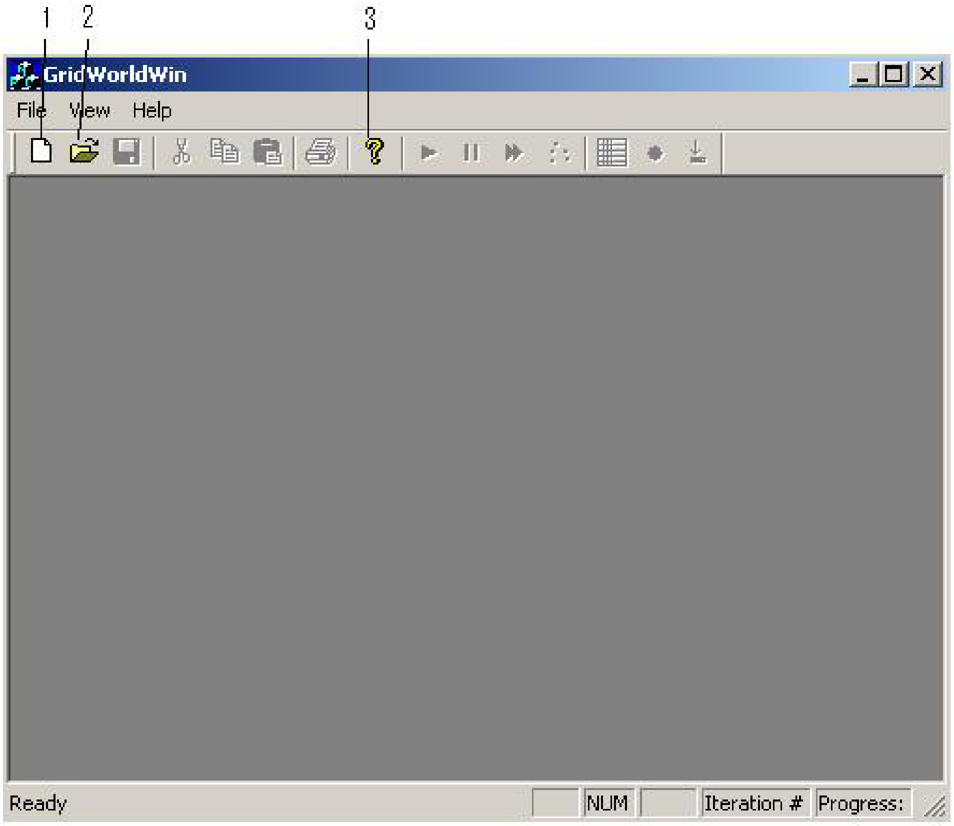


Figure A1 The mainframe of the simulation program

4. Setting up simulation configuration dialogue

Because we want to create a new simulation, we click the "create new simulation" button (button 1), and the simulation configuration dialogue will show up (see Figure A2). The simulation configuration dialogue allows users to set parameters before submitting the simulation to run.

In the simulation configuration dialogue, there are four groups of parameters contained by four group boxes. As suggested by the group box names, these four groups of parameters are for offender settings, target settings, police settings, and simulation settings respectively. Note that the parameters listed in the simulation configuration dialogue are only part of the parameter settings, which mainly focus on routine activity nodes, street network, and number of agents, number of simulation iterations, and output path.

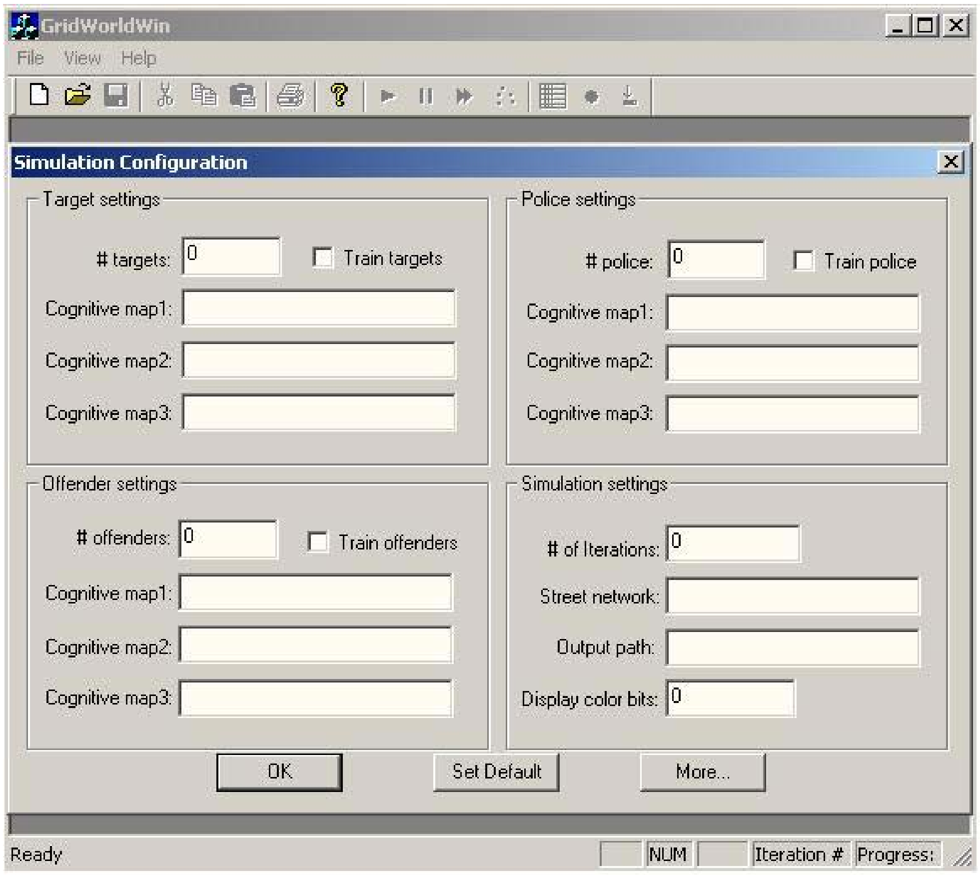


Figure A2 The simulation configuration dialogue

The “target settings” group box contains target agents number, whether include pre-training step for target agents or not, and activity nodes for target agents. In this example, we put “100” as target agents number in the edit box following “# targets:” label; check the “train targets” box, so that target agents have some spatial knowledge before the simulation actually start to run. And, because target agents have three types of activity nodes, we use each layer of activity nodes to initialize target agent cognitive map. Therefore, in the edit box following “Cognitive map1:”, we put “\GridWorldWin\Data\home”; in the edit box following “Cognitive map2:”, we put

“\GridWorldWin\Data\coevolve” (coevolve is grid representation of two shops); and in the edit box following “Cognitive map3:”, we put “\GridWorldWin\Data\work”.

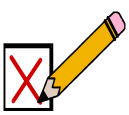
The “offender settings” group box contains offender numbers and offender routine activity nodes. For offender numbers, we put “20” in the edit box following “# of offenders”; for the same reason as target agents, we check the “train offenders” box.

Setting offender routine activity nodes requires more consideration. In this example, offender agents only have two activities, random walk and go shopping (how to represent this in an activity schedule will be discussed later). Random walk is an activity without any nodes for agents, and agents don’t need any cognitive map to direct random movement. Therefore, for random walk activity, no activity nodes need to be specified. The only activity nodes that offender agents need is the shops. The routine activity schedule is made in such a way that the shopping activity / shopping nodes are associated with the second cognitive map in an offender agent’s memory. Therefore, we put “\GridWorldWin\Data\null_grid” in the edit box following “Cognitive map1:” and the edit box following “Cognitive map3:” (null_grid is a grid in which all cells are in NO_DATA value, this is used when an activity type is not used by agent), and put “\GridWorldWin\Data\coevolve” in the edit box for the second cognitive map.

For police settings, because we will not include any police agents in this example, we can just put “0” in the “# police” edit box, and leave other boxes empty.

The last group of parameters in the simulation configuration dialogue is the “simulation settings” group. In this group of parameters, we can set the length of the simulation in terms of number of iterations; the street network used in the simulation; the output path of simulation results; and system display color bit setting. In this example, we set the number of simulation iterations to be “100,000”, which is put in the edit box following label “# of iterations:”. In the edit box following “Street network:”, we put “\GridWorldWin\Data\aoi” (this is a street network data coming with this software). The output path is set to “C:\Temp\”, so that the crime event log and crime spatial pattern will be written to this directory when the simulation finishes (The name of the crime event log file will be “log.txt”, and the crime pattern file will be “pattern.flt”. However, the file names will include the string after the last slash in the output path specified by the user, if there are any. For example, if the user put “C:\Temp\ww” as the output path, then the crime event log file will be named “ww_log.txt”, and the crime pattern file will be named ww_pattern.flt. This allows users to use different names for different simulations to save output).

Finally, there is an item in the “Simulation settings” group – “Display color bits” setting. It is important to have this value set correct to make the display work appropriately. This value should be set according to system display color setting. If the system display is set to “True Color (32 bit)”, then we should put “32” in the edit box. Or, if the system display is set to “High Color (16 bit)”, then we should put “16” in the edit box.



Warning: Inappropriate setting of the display color bits will disable the video display of the simulation.

5. Setting up agent property dialogue

The simulation configuration dialogue is in charge of settings that are not supposed to be changed by users during the simulation, such as agent routine activity nodes and street network. There are many other parameters also need to be set before the simulation can be started, such as agent routine activity schedules. These parameters are set by agent property settings dialogue (see Figure A3). The agent property settings dialogue can be opened by clicking the button

on the simulation configuration dialogue. The same dialogue can be opened when the simulation is running, which allows the users to change parameters during runtime.

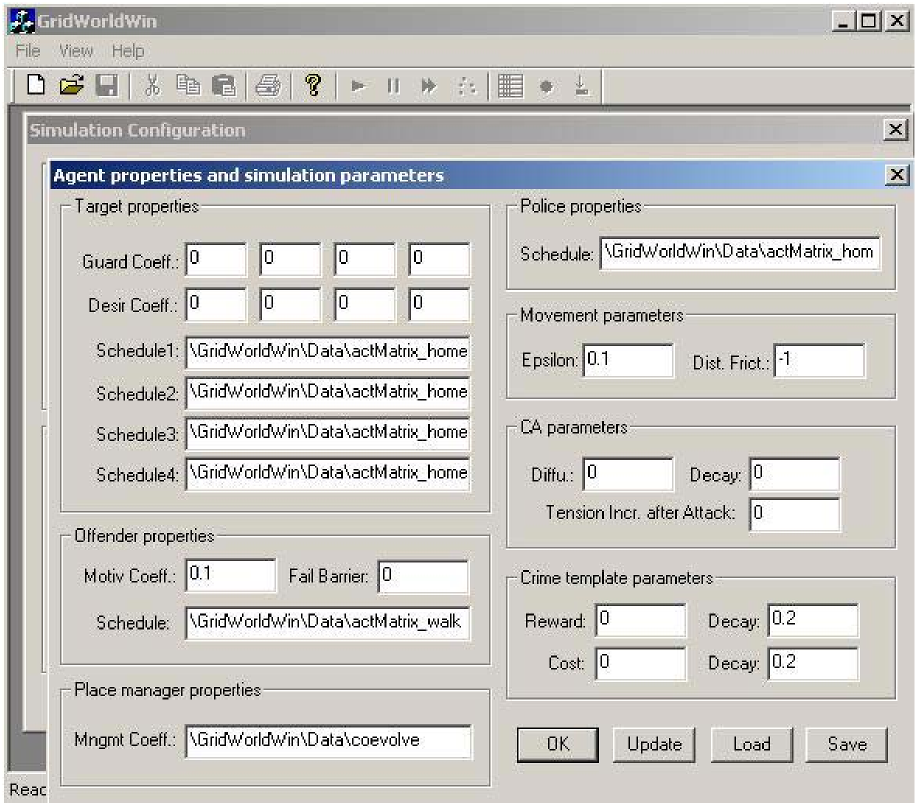


Figure A3 The agent property settings dialogue

The parameters in agent property dialogue are grouped into seven groups. The target properties group contains three parameters for target agents: guardian capability coefficient, desirability coefficient, and target routine activity schedule. Target agents are divided into four groups, each group have their own guardian capability coefficient, desirability coefficient, and routine activity schedule. There are four edit boxes following the label “Guard Coeff.:”, the first box is the guardian capability coefficient for group 1 targets; the second box is the guardian capability coefficient for group 2 targets, and so on. Similarly, the first edit box following the label “Desir Coeff.:” is for group 1 targets desirability coefficient; the second edit box is for group 2 targets desirability coefficient, and so on. The edit box following “Schedule1:” is the routine activity schedule for group 1 targets, the edit box following "Schedule2:" is the routine activity schedule for group2 targets, and so on.

In this example, we don't need to differentiate different groups of targets. Therefore, all targets are set with the same parameters. We accept the default value "0" as the setting for all guardian capability coefficients and all desirability coefficients. The routine activity schedules are all set to be "\GridWorldWin\Data\actMatrix_home_groc_work\" (which is the routine activity schedule for the routine from bus stop→shop→work place) for the four groups of targets.

For offender properties group, there are three parameters: motivation coefficient, crime event failure barrier, and offenders routine activity schedule. In this example, we set crime event failure barrier to be 0 (so that no crime event will fail), motivation coefficient to be 0.1, and set the offender activity schedule as "\GridWorldWin\Data\actMatrix_walk_groceries\".

For place manager properties group, there is only one parameter – management effectiveness (the edit box labeled by "Mngmt. Coeff.:"). This parameter needs a grid data layer storing the initial values of management effectiveness for all places. In this example, we use "\GridWorldWin\Data\coevolve" as the management effectiveness initial values, and put it in the "Mngmt. Coeff.:" box.

For police properties group, there is only one parameter – the routine activity schedule for police agents. In this example, because police agents are not considered, it does not matter which activity schedule is used (for example, we can use the same schedule as offender agents for police agents, and put "\GridWorldWin\Data\actMatrix_walk_groceries\" here).

The next parameter group is movement parameters. The "Epsilon:" parameter is for setting the epsilon-greedy action of agent movement on the street network. This value is set to be 0.1

under all cases. The "Dist. Frict.:" parameter is to set distance friction. The higher the distance friction, the less mobility/adaptability agents possess. This value is set to be –1 in this example.

The "CA parameters" group sets the cellular automata parameters. "Diffu.:" represents tension spatial diffusion coefficient, and "Decay:" represents tension temporal decay coefficient. "Tension Incr. after Attack" represents tension increase after each crime event at the crime place. In this example, cellular automata simulation of tension is not necessary, so we set all these parameters to "0".

The last group of parameters are crime template parameters. Crime templates are memories for offender agents and target agents to store the reward received from the environment. "Reward:" represents the reward offender receives from environment following each crime. The "Decay:" following "Reward:" is the temporal decay coefficient of offender reward. "Cost:" represents the reward target receives from environment following each crime. The "Decay" following "Cost" is temporal decay coefficient of target cost. If "Reward" and "Cost" are set to zero, then offender and target could not perceive the change of environment, and will not be able to adapt to new activity patterns. In this example, these two parameters will be set to different value to test their effect on offender and target agents adaptability. First, let's set them both to "0", so that offender and target agents have no adaptability initially.

<u>6. Starting the simulation</u>

After step 5, all parameter settings are set up and the simulation is ready to get started. Click "OK" button on agent property settings dialogue (Figure A3), the program will return to simulation configuration dialogue. Next, click "OK" button on the simulation configuration dialogue, simulation will be loaded. After the simulation is loaded, the video display window will open. Now, the interface is ready to run. At this moment, the interface is illustrated in Figure A4.

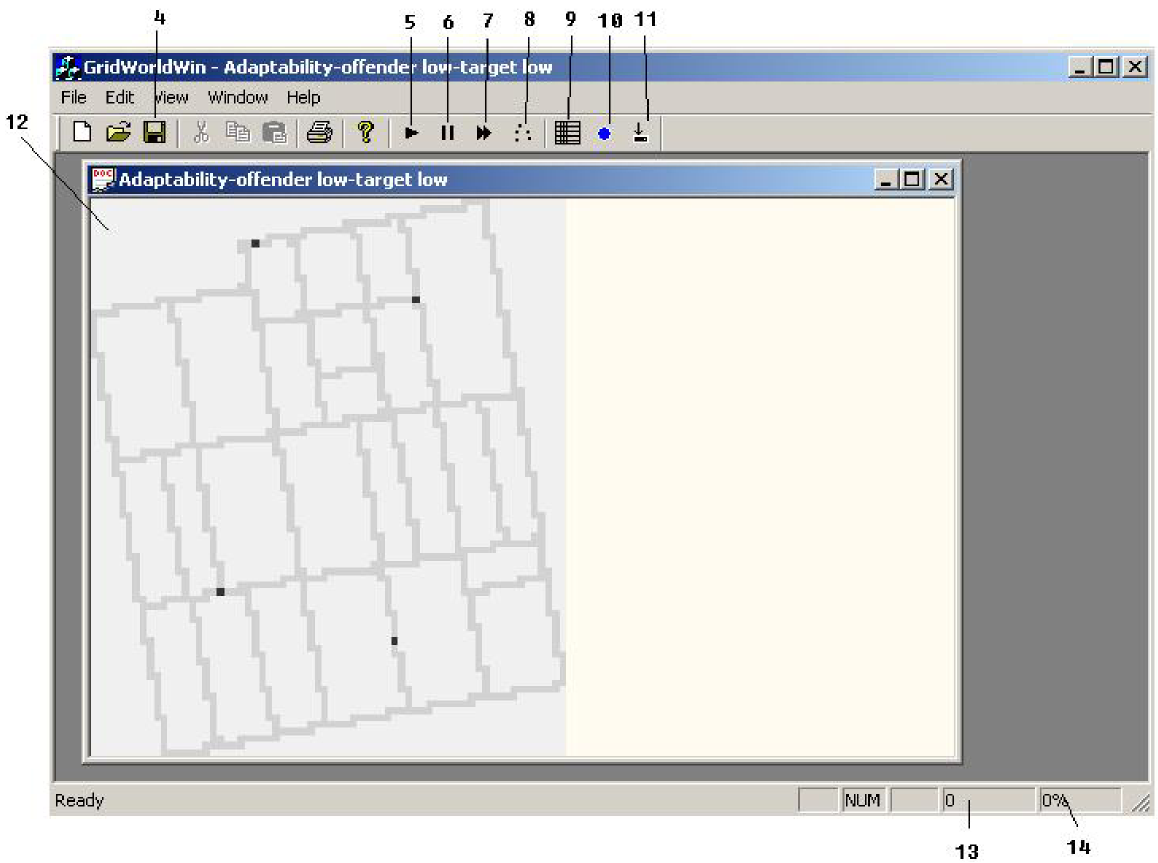


Figure A4 The simulation program interface after the simulation is loaded

Compared with Figure A1, the interface now shows more controls. First of all, buttons 4, 5, 6, 7, 8, 9, 10, and 11 become available. A window is also created (item 12 in Figure A4), showing the street network and activity nodes (the black dots).

To start the simulation, click button 5. After the simulation process is started, offender and target agents will be shown on the display window (in red and blue colors respectively) (see Figure A5).

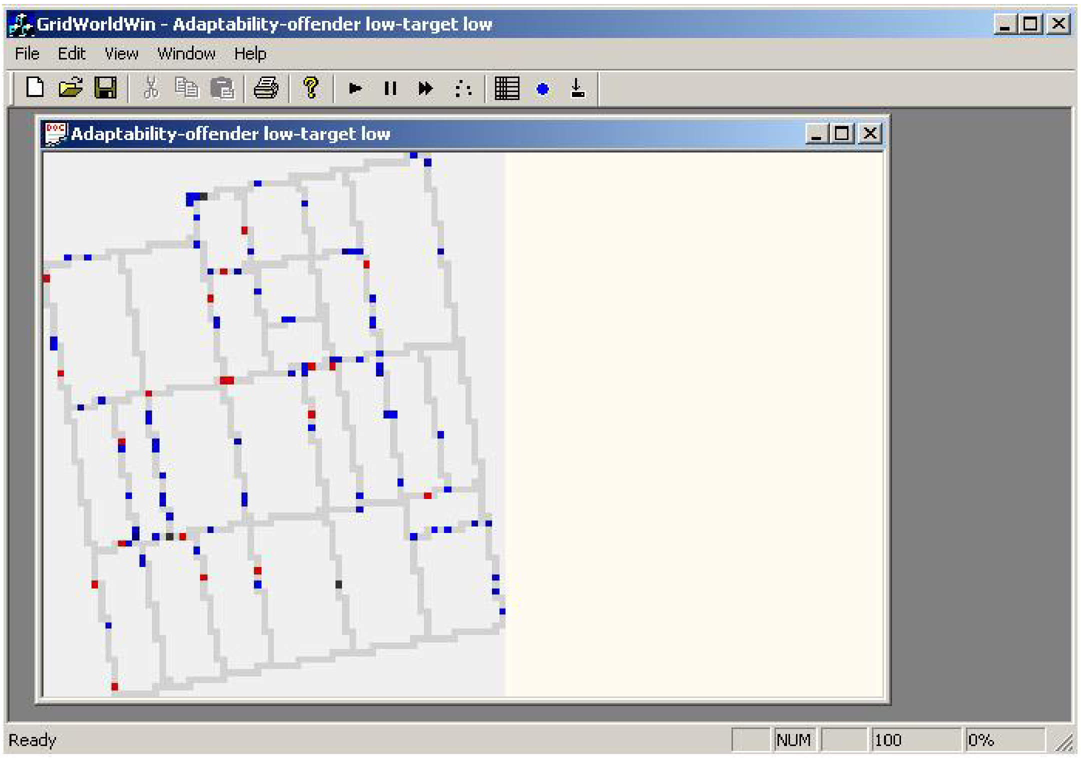


Figure A5 A snapshot of the video display of the simulation process (red is offender, blue is target)

7. Saving parameter settings

To save the configuration for this simulation, click button 4, or click "Save As" under the "File" menu item.

8. Pause a simulation

To pause a simulation process, click button 6. When the simulation pause, the display window will also freeze. To resume a simulation, one can click button 5 or button 7.

9. Turn the simulation into a fast mode

Clicking button 7 will turn the simulation into a fast-run mode. In fast mode, the video display is not available. To resume the video display, click button 5.

10. Show crime pattern

To show current accumulated crime spatial pattern, click button 8. No matter the simulation is in normal running mode or fast mode, clicking button 8 will always show crime pattern. To return to the display of agents, click button 5.

11. Change parameters during runtime

Agent properties and some of the simulation parameters can be changed during runtime. In order to change parameters during runtime, we click button 9. Then the agent property settings dialogue will show up (the same dialogue shown in Figure A3). Users can change any parameters listed in the dialogue, and then submit these changes through clicking either the “Update” button or the “OK” button on the agent property settings dialogue. The difference between the two buttons is that “Update” button will leave the dialogue visible, but “OK” button will make the dialogue disappear.

12. Setting break points for the simulation

Users can pause the simulation at any time manually (through clicking button 6). Or, users can set up regular break points for a simulation, so that the simulation pauses at regular intervals. For example, if the user set 100 as the interval of each break, then after every 100 iterations, the simulation will pause automatically. This feature is especially useful when the user needs to make observation during the simulation at regular intervals. Manually pausing the simulation is difficult to make it at desirable points. To resume the simulation, users need to click button 5 or button 7.

To set up the break points, the user needs to click button 10. After clicking button 10, a small dialogue will show up (see Figure A6). If the user wants the simulation to pause every 100 iterations, then the user needs to put “100” in the edit box following “Pause every” and before “iterations”. At the top of the dialogue is a check box labeled as “Enable automatic pause”, the user MUST check this check box in order to make the automatic pause function work.

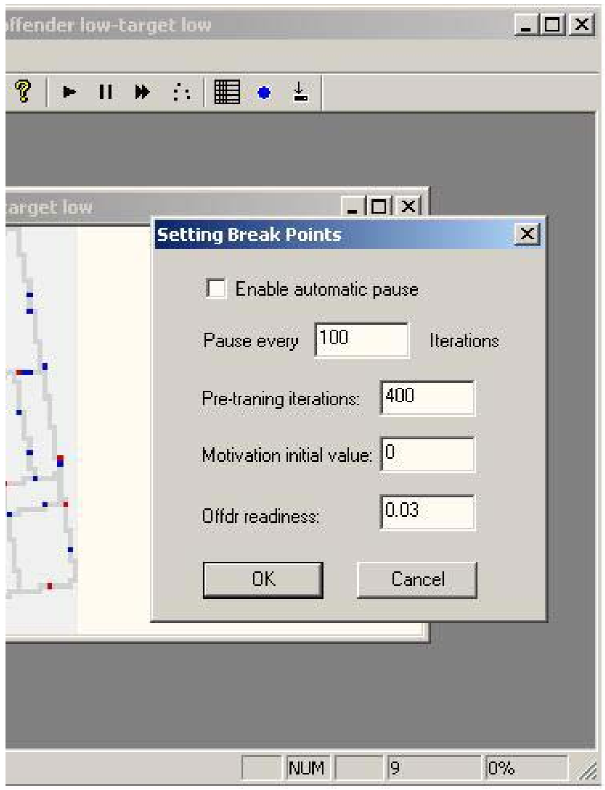


Figure A6 The dialogue for setting simulation break points

13. Set and change motivation initial value

During the simulation, users can control the motivation initial value through the breaking point setting dialogue in Figure A6. In this example, we first set the motivation initial value to be 0, and after iteration #1500, we change it to 3.0. Then we can observe the change of agent routine activity patterns.

In order to set offender motivation initial value, the user needs to put the value into the edit box following “Motivation initial value:” in the breaking point setting dialogue.

14. Monitoring the progress of the simulation

During the simulation, users will need to know the progress of the simulation. This information is available on the interface of the simulation program. Item 13 in Figure A4 is the indicator for current iteration number; item 14 is the indicator for current progress (percentage of iterations finished among total number of iterations).

15. Query agent properties and crime spatial patterns

In some studies (such as motivation development), agent properties need to be diagnosed over time. The dynamic development of crime spatial patterns is also interesting. The simulation program provides an interface for users to diagnose agent properties and save these properties in ASCII files.

Currently, two types of properties are available to be queried during simulation: offender motivation values and crime spatial patterns. When the user clicks button 11 in Figure A4, these two properties will be exported to local disk according to the path and name specified by the user.

IV. Opening existing simulation

The simulation parameters set by the three dialogues (Figure A2, A3, and A6) can be saved on a local drive by clicking button 4 in Figure A4, or clicking the “Save As” menu item under the “File” menu. The advantage of the saving function is that it allows users to save each simulation configuration, so that users don’t have to create new simulations each time. As illustrated in the “creating new simulation” section, creating new simulation is a time-consuming task.

As examples, the software package that contains this program already includes some simulation configuration files created by our development team. Users can open these simulations directly to run them. In order to open existing simulation configuration files, the user needs to click button 2 of Figure A1 and choose the correct file name in the dialogue.

The “Open existing simulation” function is also useful for users in creating new simulation. The user can open an existing simulation configuration file, and modify the parameter settings for a new simulation, and then save the modified simulation configuration as a new simulation configuration file. This can save a lot of efforts in creating new simulations.

V. Preparing spatial data

The spatial data used in the simulation include street network and routine activity nodes of agents. These are all in grid format. This software package already contains some spatial data used as examples. Users can directly use these data, or they can create their own street network and routine activity nodes.

1. Software needed in preparing spatial data

ESRI Arcview3.2 with Spatial Analyst extension is the GIS software used in preparing street network and routine activity nodes.

2. Procedures in preparing street network

Start Arcview3.2, and add the street network line data into a view of the project. Highlight the street network theme, and click the menu item “Convert to Grid” under “Theme” menu. This operation will convert the street network vector data into raster data. Remember the path and name of the street network grid.

Next, the user needs to export the street network grid into the format that can be read by our simulation program. Click “File” menu and pick “Export data source” (the Spatial Analyst MUST be activated for this menu item to be available). In the following file dialogue, choose the grid to be exported. Then, specify the format of output grid data as “Binary raster”. The output grid file will have an extension name as “.flt”, which is the format used by our simulation program.

3. Procedures in preparing activity nodes

Activity nodes are destinations of agent daily travel, such as home, work, school, entertainment, etc. These activity nodes need to be converted into grid to be used by our simulation software. There are two steps in making a routine activity node layer (note that each activity node layer can include several points, for example, many shops can be put into one layer):

Step1: Create the vector point layer

First, the user needs to put the coordinates of the nodes in a text file, and then add the text file as a table to the Arcview project. Second, use the "Add event theme" menu item under "View" menu to convert the table into a vector point layer. Add a numerical field "ManageK" to the table . Third, initialize the "ManageK" field with the desired management effectiveness value.

Step2: Convert the vector point layer into grid. The grid stores values of the "ManageK" field. Export the grid as ".flt" format so that it can be used by our simulation software. It is recommended that users put the final street network and activity node files in the directory of "\GridWorldWin\data\".

VI. Preparing routine activity schedules

There are several routine activity schedules prepared by the development team included in this software package, which can be directly used in simulations. Users can also create new schedules.

1. What is a routine activity schedule?

Each routine activity schedule is a directory containing 8 text files (named from “0.txt” to “7.txt”). Each text file is a transition matrix of agent activities for 1/8 of a day. For example, “0.txt” is the transitional probabilities of agent activities for 0:00-3:00 am.

Each matrix of transitional probabilities includes 6 activities, which means that agents can be designed with 6 activities at most. Table A1 is the transition matrix for target activities in this example (which can be found under directory: \GridWorldWin\Data\actMatrix_home_groc_work\):

Table A1 Transition matrix for targets

Activity1\activity2	0	1	2	3	4	5
0(reserved)	0	0	0	0	0	100
1(reserved)	0	0	0	0	0	100
2(shop)	0	0	0	0	100	0
3(bus stop)	0	0	100	0	0	0
4(work place)	0	0	0	100	0	0
5(rand. walk)	0	0	0	100	0	0

In every transition matrix, there are six entries (labeled from 0 to 5 at the beginning of each row). Each entry represents a type of activity. In the examples coming with the simulation program, the definitions of the six activities are as follows:

0 – reserved for future development use;

1 – reserved for future development use;

2 – going to shopping locations;

3 – going to bus stop;

4 – going to work place;

5 – random movement;

Each column is also a type of activity, with the same sequence as rows. Each cell of the table stores the transitional probability from activity1 to activity2. For example, the value at row3 and column2 is 100, meaning that the transitional probability from bus stop to shopping location is 100%. The value at row3 and column4 is 0, meaning that the transitional probability from bus stop to work place is 0%. Other values in the table can be similarly interpreted. This transitional probabilities matrix schedules that targets move from bus stop to shop location to work place, and loop.

Similarly, Table A2 shows the transition matrix that schedules offender agents "wander around" the two shops:

Table A2 The transition matrix for offender agents

Activity1\Activity2	0	1	2	3	4	5
0(reserved)	0	0	0	0	0	100
1(reserved)	0	0	0	0	0	100
2(shop)	0	0	0	0	0	100
3(bus stop)	0	0	0	0	0	100
4(work place)	0	0	0	0	0	100
5(rand. walk)	0	0	1	0	0	99

In Table A2, the value at row5 and column2 is 1, meaning that the transitional probability from random movement to shopping is 1%. Most of the time the offender will be in a random movement status.

2. How to set up a routine activity schedule?

To create a new routine activity schedule, the user first need to create a folder to hold the eight transitional probabilities matrixes. The path of the folder on local drive is provided to the simulation program, so that the simulation program can read the transition matrixes. For each of the eight transition matrixes, the user can use Table A1, or Table A2 as a template to set up the matrix. These eight transition matrixes can be the same, or they can be different. If they are different, then it means that agent activities are different at different time interval of the day.