Computer Science and Engineering
Jahangirnagar University

A research project submitted to the Department of Computer Science and Engineering, Jahangirnagar University in partial fulfillment of the requirments for the degree of

B.Sc in Computer Science and Engineering

Spatio-temporal analysis of agent-based bicycle traffic model

---

Submitted By :

Class Roll : 26
Exam Roll : 150026

Class Roll : 21
Exam Roll : 150021

Date : July 31, 2019

Contents

List of Figures

Abstract

Bicycle sharing systems are emanating as secondary injunction form of transportation now a days. Jobike is an on demand bike sharing service which gives everyone the opportunity to make short trips easily within the coverage area. Spatio-temporal models arise when data are collected across time as well as space and has at least one spatial and one temporal property. Here we show the spatio-temporal analysis of bicycle system based on agent-based modeling. In ABM multiple agents are interacting. Cyclists use the jobike when they need. For attaining cycles in their needs, an initial prediction helps them by knowing the existance of cycles which is done by autonomous agent. This analysis gives a clear view about jobike distribution in specific area. As a result the bicycle availability is increased. Using spatio-temporal analysis of Jobike movement finishes with the result of bicycle distribution in various place at the end of the day.

Chapter 1

Introduction

1.1 Overview

Agent based modeling has demandable usage in research work for simulation. Social scientists and demographers are interested in agent-based modeling and simulation for solving many socio-economic problem. In municipal area bicycles play an increasingly significant role in maintaining traffic strategies.

Recent research contributes to a better understanding of the spatio-temporal analysis of cycle trips and emerging cycle traffic flow patterns. The traffic demand model is the spatial flows and distribution of bicycle traffic with operable model [Cascetta 2001]. Spatio-temporal bicycle model illustrates about where and when how many cycles are on trip. It also bears the significance about cyclists behaviours in different application context in traffic management. Here, we develop an agent based model for cycle traffic in Jahangirnagar University campus area and validate the model with available counting data. We see that for the casual movement of jobike. It is scattered in differnt places. Consequently, maximum jobikes are appeared at specific station and the other places are empty. The availability of the jobike is hampered. The scarcity of data on bicycle trip patterns thus impedes sound analysis. So from the simulation we make a prediction of availability of jobike. Such that The autonomous agent collects jobike from predicted area then distribute these in predefined station.

Agent-based modeling and simulation composed of heterogeneous interacting agents, with several features which turn them into a significantly winsome modeling approach to simulate intricate social systems. Simulation results are affirmed against data from cycle stations. To know when and where how many and which cyclists are on the road is of illustrious importance for a diversity of application contexts from infrastructure planning to traffic management.

Here, we develop an agent-based model for cycle traffic in Jahangirnagar University campus area and validate the model with available counting data. We see that for the haphazard movement of jobike it's scattered in differnt places. Consequently, maximum jobike is appeared at specific station and the other places are empty. The availability of the jobike is hampered. The scarcity of data on bicycle trip patterns thus hampers sound analysis and consequently a better understanding of appointments behind bicycle usage [Vandenbulcke 2011]. So from the simulation we make a prediction of availability of jobike. Such that The gleaner agent collect jobike from predicted area then distribute these in predefined station.

This research utilizes open-source Netlogo 6.0 software [Wilensky 1999a]. Using Netlogo and empirical data , this research follows fundamental tenets of traditional and contemporary mathematical modeling theory to identify the large-scale impact of asymptomatic carriers caused by jobike and customer interaction. Our proposed model is properly adjusted to physical movement of jobike bicycle traffic system.

1.2 Motivation

At the present time, the understanding of complex works or solving a socio economic problems , rapid enumeration is used through modeling and simulation of a system.Models attempt to represent and simulate real or imagined scenarios, which we call the target. These models are generated as artificial view of a system.Simulation helps to solve the real world problem .For Traffic analysis here Agent Based Modeling is used.Agent Based Models are computer models that attempt to capture the behaviour of individuals within an environment. Spatio temporal analysis of bicycle model is the point of our research topic.

Many types of ecological, environmental data are collected over discrete spatial and temporal domains. Many social phenomena have a spatio-temporal dimension and involve dynamic decisions made by individuals. The jobike bicycle model reflects our current understanding of cycle traffic in Jahangirnagar University campus. In campus there exists a bike sharing system called jobike. Students, working people and other people canb use it as riding . They make trip in the campus area. So, at the end of the day the Bicycles are spreaded here and there in the whole area of campus. Jobike management body collect all the cycles at late night and distribute these to several places of campus. If the collector can know the availability of the cycles,he can collect cycles easily.

So the spatio temporal analysis of bicycles result in a time and place specific distribution of it.Agent based model is used to generate the distribution of cycles.

1.3 About ABM

The new age of expert technology has occupied many complex problems and works on to solve these through complex models. Agent-based modeling creates suitable scope of accomplishment of complex problem solving in engineered, social, and natural contexts. Agent-based modeling is an alternative way of completion a system modeling which is a substitution of equation based modeling [Singh 2011].

In ABM a system is modeled as a set of self-explanatory agents who interact with each other as well as with the environment. The agents represent various actors in the system and sense the environments (Fig. 1.1).

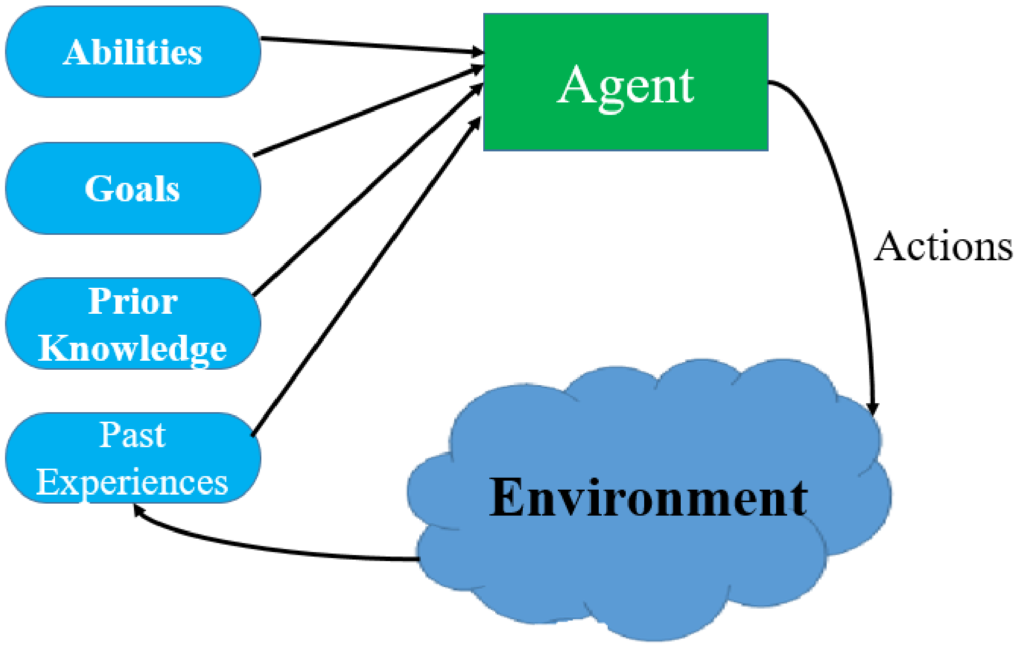

Figure 1.1: Agent based model.

The environment represents agent's surroundings and the overall context. Agent based modeling has been successfully applied to model and analyze numerous complex social processes. In a system composed of multi-agent, each agent is autonomous and independent. It is a type of software abstraction that describes a complex software entity efficiently. An agent is defined in terms of its traits and attributes. Each agent is intelligent, has only local knowledge and maximize its own interests. Agents resolve the conflicts and disharmony by means of negotiations. Its own actions as well as objectives don't be restricted by other agents.

The term "agent" is used in a variety of ways in cognitive science and computer engineering. As the term is used in modeling, an agent tends to have a number of characteristics, although a range of variability exists on many of these dimensions [Smith 2007]. Agent-based modeling and simulation techniques have now become a suitable. Agent-based models consent simulating social units such as individuals, households, organizations or nations and their direct or indirect interactions.

The Agent-based modeling approach has provided the bridging link between psychological sociological analysis of individual and social behaviours respectively, which was otherwise missing [Singh 2009]. It is a popularly used approach to build useful models of social systems, which not only helps to get better understanding of various social phenomena. These models demonstrate how global order and collective intelligence can emerge from relatively simple local interactions and can explain the dynamics of the emergent behavior. This generative, evidence by construction approach has also complemented the individual centered research in cognitive science by showing that individual alone is not the conclusive unit of cognition but is affected by environment and society besides affecting them as well. In this paper, we have given an analytical account of agent-based modeling of exigent collective social conducts, on these lines, along with pertinent theoretical experimental outcomes and their implications for multi-agent.

1.4 Why we need ABM

What we see immediately is that ABM has pretty good coverage across all potential system features. Where it has no coverage, that's because it would be a waste of time to build an ABM given the availability of other techniques. If we're trying to model the mean field of a system, there is no point building out an ABM only to go back to recover the mean of the agent behaviour.

We also see is that there are some features that ABM covers that few other approaches cover, such as local and spatial interactions, and individual level behaviour. If we need an individual level model with spatial interactions, ABM is a good way to go.

An agent can represents anything, from a node in a network to an agent or a system. The idea is to disentangle the properties and behavioral rules of these agents in their specific setting, and then run multiple simulations to understand the central tendency of the system. Learning agent-based modeling is often difficult, but if well-introduced with hands-on exercises during an intensive training, everyone can learn how to perform advanced computational research that matters [sandtable 2017]. Given below the importance of agent-based modeling :

- ABMs are extensible: models can be incrementally advanced and refashioned to answer new business questions.
- ABMs are interpretable. For example, they support transparent and granular sce-

nario planning based on real-world concepts.

- ABM is an holistic modelling approach that permits multiple questions to be answered across a complete system (i.e. ABMs are many question models). In this sense, it can be used to reconcile two contradictory models of different parts of the system.

- It's often the case that individual-level behaviour is better understood than aggregate (population) level. Since it works at individual-level, ABM starts with something that is often better understood.

- ABM embraces complexity: as such, it doesn't require the introduction, early in the modelling process, of a host of simplifying assumptions.

- Finally, there's a subject bias for ABMs: they are particularly well suited to modelling behaviour, because of their reliance on the agent and agency metaphor.

An ABM, not only considers the external factors, but also one component's in- teraction with other components, to decide their actions. Thus, it includes the possibilities of these interactions impacting the actions as well. Also, as stated by William Rand, Consumers modeled with ABM can be boundedly rational, heterogeneous in their properties and actions, adaptive and sensitive to history in their decisions, and located within social networks or geographical locations. This makes simulations of situations using ABMs more representative of the real world.

1.5 Document Layout

In chapter one, we introduce the project. Here we describe what agent-based modeling is and why we need this agent-based model. Also introduce the overview of the model.

In chapter two, we describe agent-based modeling history and the authorizing tools used for creating agent-based model. We explain the component and perception of agent-based model. Also we illustrate the spatio-temporal analysis and it's data workflow.

In chapter three, contains the description of the proposed model and it's architecture. Also we explain here the purpose and agents of the proposed model.

In chapter four, we describe the implementation of the system. The coding for procedures. The initialization and input data also explained here.

In chapter five, contains result of the model and validation of the model.

In chapter six, contains the summary of the system.

Chapter 2

Literature Review

2.1 History of ABM

Agent-based modelling is only a fashionable new name for the use of computer simulations in economics, which exists almost as long as computers. When in the fifties and sixties large US universities installed computer centres to support researchers in the natural sciences, there soon were also economists showing up, eager to transfer tedious calculations to the new machinery. Before that, large amounts of data as well as larger difference equation systems built on the the new analytical technique of input-output tables were using up a lot of intellectual effort that now could be delegated to computers. The advent of computer simulation therefore can be considered as the appearance of 'an economists little helper', a calculating machine.

Then came a shift of focus that was initiated by the computer firms acting in anoligopoly, and that quickly was taken up by the scientific interpreters of the new machines. The growth of demand for computers coming from military and other state institutions was starting to slow down, and to revive it in the 'age of high mass consumption' [Rostow 1960] made smaller and more flexible devices necessary. Small firms and households did not need extreme calculation power, like research in theoretical high-energy physics. What they would buy was a small machine that offered a large diversity of different uses, from simple household accounting via writing letters to computer games for children and idle employees. In the early eighties, the personal computer was born, and as an educational add-on the teaching of informatics in schools was introduced. Scientists like Alan Newell and Herbert Simon quickly saw that with this diversity of uses the old calculator had become a symbol manipulating device [Newell 1972]. Indeed, whatever one can consider as a symbol can easily be translated in a bit stream and then can be manipulated in many more ways than just adding and subtracting. At that time another fashionable term was invented to attract research money: 'artificial intelligence'. Its success was built on the surprise of computer illiterates about the speed of machines searching in databases. This looked 'intelligent', though it had little to do with the traditional meaning of the word3 . Till today 'artificial intelligence' haunts software applications, including economic simulations.

The early fields of application quite naturally followed the divide of economics into microeconomics and macroeconomics. On the one hand, simple oligopolistic markets were mirrored, while on the other hand early Keynesian macroeconomic models were simulated. Of course, contrary to abstract general form models of economists these models first had to be econometrically estimated to fill in parameter values into the behavioural equations. Without using the word 'agent' such a behavioural equation already was meant to capture the behaviour of an economic entity, of economic agents. In microeconomics they were firms, households, or individuals, in macroeconomics they were more abstract aggregates of 'all households', 'all firms', or 'several state institutions'. With the increasing technical abilities of hard- and software soon there emerged first attempts to combine micro- and macroeconomic computer simulations, a task paralleling the more conventional efforts of mainstream economic theory to provide a microfoundation for macroeconomics. In the late eighties, it more or less became clear that the latter task had failed since it needed very implausible restrictions on the micro-level to allow for aggregation at all. Moreover, the convenient trick only to consider stable equilibria, or equilibrium paths, became less and less useful in a world that was governed by amplifying disequilibria, in the labour market. But the human capital stock of academic economists built up by this type of research was already too large to be easily given away. The alternative to use computer simulations would have needed too much investment in learning the new techniques for the established community, it became a territory for mavericks only. .

2.2 Perception of ABM

The exposition of extensive fast computing has enabled us to work on more complex problems and to build and analyze more complex models. The compatibility of different agent-based modeling software tools for bicycle flow modeling, here it is identified a set of evaluation criteria and construct an evaluation scheme [Kaziyeva 2018]. The flowing model of bicycle is based on these specific outlines. The selection of these standards was based on the requirements of bicycle flow modeling [Abar 2017]. These criteria are briefly discussed below :

- **Platform intention :**

  Ensures general and domain-specific functionalities and pliability.

- **Licence :**

  Ensures presence, debonair and extensibility.

- **Model uplift effort :**

  An ultimate feasibility factor in terms of the effort needed for developing a new model.

- **Integrated development environment (IDE) :**

  Provides functionalities of coding, debugging and running models.

- **Performance :**

  High performance is particularly important in transport models to represent an agent's complex, adaptive behaviour during daily activities, their choice of transport mode, and route preferences during a simulation. This micro-level behaviour of individual cyclists depends on the one hand on the agent's response to environmental situations, and on the other hand on the presence and behaviour of other road users.

- **Spatial capabilities :**

  Ensure the implementation of cycling behaviour, which requires spatially explicit representation of networks .

- **Output visualization :**

  Facilitates verification of output and understanding of model behaviour.

- **Platform maintenance :**

  Regular software updates beyond mere bug-fixing, that constantly augment functionality and usability

2.3 Component of ABM

An agent-based model consists of following core components. The description of the components are given below :

- **Agents :**

  Agents can represent a person, but also an animal, or an organization. To decide if something is an agent we need to check the agent characteristics. These are characteristics most agents have. They range from "a goal", "exist within the environment" to "communication" and "movement". On organization may not have a location or cannot move, but when it has a goal, and behavior that it can execute to reach this goal this is enough to identify it as an agent. Behavior can be implemented via simple rules. These rules consist of a condition (IF) and the behavior to execute (THEN). For example If the number of disease cases shows a rise then start a vaccination campaign [badu osei]. Social things in a model that can interact with each other and an environment, and pass information between each other. Agents might represent animals, individuals, households, organisations, or even entire countries.

- **Properties :**

  Agents have properties. A property might be memory, a state, characteristic, or attribute, such as hunger, speed, or health. Properties are discrete, and can be binary (yes/no), or numerical.

- **Environment :**

  Environments are the space in which agents exist. An environment can be abstract (empty space) or can be a layer from GIS representing true geographic information. Most simulations have multiple environments. Environments can be either static or dynamic. When an environment is dynamic, it will change over time. For example, river water may contain cholera during a certain period of simulation run, but it may also be safe to drink during other time steps. This is the virtual world in which agents act and interact. It can be 2D, or 3D. A neutral medium with no effect on agents whatsoever, or a prime determinant of their ability to act. It can be abstract and imagined, or a replication of real-world buildings, cities, or vast geographies.

- **Rules :**

  These govern what happens when agents interact with each other or their environment. They may also govern how learning and adaptation occur within an environment [Wilkinson 2018]. These rules may be pre-programmed or automatically inferred in the case of some smart ABM platforms.

- **Time**

  Time is the third component of a simulation. A simulation will go through a number of time steps (ticks). It can be that each agents and dynamic environments will be updated one time during every tick. For example, during every tick the disease can be transferred to other agents. A tick can have different durations just like the complete simulation. A tick can represent a second, a day or a year. Depending on the duration of the tick, different behavior can be implemented.

2.4 Agent-based modeling tools

The agent-based modeling (ABM) community has developed several practical agent based modeling toolkits that enable individuals to develop agent-based applications. More and more such toolkits are coming into existence, and each toolkit has a variety of characteristics. Several individuals have made attempts to compare toolkits to each other [Nikolai 2009]. Here is a list of the current agent-based modeling simulation environments. A Survey of Various Agent Based Modeling Platforms are given below :

2.4.1 Netlogo

NetLogo is a programming language and integrated development environment (IDE) for modeling. It is agent-based.

NetLogo was designed by Uri Wilensky, in the spirit of the programming language Logo, to be "low threshold and no ceiling". It teaches programming concepts using agents in the form of turtles, patches, links and the observer. NetLogo was designed for multiple audiences in mind, in particular: teaching children in the education community, and for domain experts without a programming background to model related phenomena. Many scientific articles have been published using NetLogo.

The NetLogo environment enables exploration of emergent phenomena. It comes with an extensive models library including models in a variety of domains, such as economics, biology, physics, chemistry, psychology, system dynamics. NetLogo allows exploration by modifying switches, sliders, choosers, inputs, and other interface elements. Beyond exploring, NetLogo allows authoring new models and modifying extant models.

NetLogo is open source and freely available from the NetLogo website.[6] It is in use in a wide variety of educational contexts from elementary school to graduate school. Many teachers make use of NetLogo in their curricula.

NetLogo was designed and authored by Uri Wilensky, director of Northwestern University's Center for Connected Learning and Computer-Based Modeling (CCL) [Wilensky 1999b].

2.4.2 Gama

GAMA is a simulation platform, which aims at providing field experts, modellers, and computer scientists with a complete modelling and simulation development environment for building spatially explicit multi-agent simulations. It has been first developed by the Vietnamese-French research team MSI (located at IFI, Hanoi, and part of the IRD/SU International Research Unit UMMISCO) from 2007 to 2010, and is now developed by a consortium of academic and industrial partners led by UMMISCO, among which the

University of Rouen, France, the University of Toulouse 1, France, the University of Orsay, France, the University of Can Tho, Vietnam, the National University of Hanoi, EDF R&D, France, and CEA LISC, France.

Some of the features of GAMA are illustrated below :

- A complete modeling language, GAML, for modeling agents and environments.
- A large and extensible library of primitives (agent's movement, communication, mathematical functions, graphical features etc.).
- A cross-platform reproducibility of experiments and simulations.
- A powerful declarative drawing and plotting subsystem.
- A flexible user interface based on the Eclipse platform.
- A complete set of batch tools, allowing for a systematic or "intelligent" exploration of models parameters spaces.

2.4.3 MatSim

MATSim is an open-source framework to implement large-scale agent-based transport simulations, where a large number of individual, synthetic persons (so-called "agents") are simulated.

The framework is designed for large-scale scenarios, meaning that all features of the model are stripped down to efficiently handle the targeted functionality. For the network loading simulation, for example, a queue-based model is implemented, omitting very complex and computationally expensive car-following behaviour.

MATSim is typically used to simulate a single day, but multi-day applications could be implemented by minor modifications to the (open) source code of the tool.

MATSim is based on the co-evolutionary principle. Every agent repeatedly optimises its daily activity schedule while competing for space-time slots with all other agents on the transportation infrastructure. This is somewhat similar to the route assignment iterative cycle, but goes beyond route assignment by incorporating other choice dimensions like time choice, mode choice or destination choice into the iterative loop.

2.4.4 Mason

MASON is a fast discrete-event multiagent simulation library core in Java, designed to be the foundation for large custom-purpose Java simulations, and also to provide more than enough functionality for many lightweight simulation needs. MASON contains both a model library and an optional suite of visualization tools in 2D and 3D.

MASON is a joint effort between George Mason University's Evolutionary Computation Laboratory and the GMU Center for Social Complexity, and was designed by Sean Luke, Gabriel Catalin Balan, Keith Sullivan, and Liviu Panait, with help from Claudio Cioffi-Revilla, Sean Paus, Keith Sullivan, Daniel Kuebrich, Joey Harrison, and Ankur Desai.

Some of the features of MASON are illustrated below :

- 100% Java (1.3 or higher).
- Fast, portable, and fairly small.
- Models are completely independent from visualization, which can be added, removed, or changed at any time.
- Models may be checkpointed and recovered, and dynamically migrated across platforms.
- Can produce results that are identical across platforms.
- Models are self-contained and can run inside other Java frameworks and applications.
- 2D and 3D visualization.
- Can generate PNG snapshots, Quicktime movies, charts and graphs, and output data streams.

2.4.5 Ascape

Ascape is an innovative tool for developing and exploring general-purpose agent-based models. It is designed to be flexible and powerful, but also approachable, easy to use and expressive. Models can be developed in Ascape using far less code than in other tools. Ascape models are easier to explore, and profound changes to the models can be made with minimal code changes. Ascape offers a broad array of modeling and visualization tools.

A high-level framework supports complex model design, while end-user tools make it possible for non-programmers to explore many aspects of model dynamics. Ascape is written entirely in Java, and should run on any Java-enabled platform. Whether you are involved in academia, industry, or government, we hope that you will find that Ascape enables your efforts to exploit the profound insights agent-based models make possible.

Ascape is released under a BSD standard open source license and thus is free to use and redistribute. The Ascape distribution includes a number of other Open Source libraries; please see the licenses directory and individual jars for more information.

Ascape is research oriented software, while it is quite mature, apis, distribution, may continue may change over time. Direct support is not provided, forums provide a much better venue for sharing information and make the most efficient use of limited resources.

2.4.6 Flame

Flame (Flexible Large-scale Agent-based Modeling Environment) is a very general system for building detailed agent-based models, that generates highly efficient simulation software. It can be run on any computing platform in particular. Also it can be run directly on High Performance parallel supercomputers (HPC), as far as we know this is the only framework with this capability. It is also available in a GPU (graphics processing unit) version. It can also be run on any desktop or laptop on Macs, Windows or Linux. It is possible to set up large scale models with millions of agents. Additional FLAME documents can be found here, and more information at CCPForage and at FLAME GPU.

2.5 Bicycle traffic simulation

Bicycle traffic simulation means the mathematical modeling which create a visualization of artificial traffic system. It is used on making design, plan and decision. The simulation result is constructed based on the deposition of cyclists. The exclusive usage of broad data the approach is transferable to other spatial contexts.

Here the simulated system is made on any ABM tools, as like netlogo. The netlogo world is made up of four agents which are turtles, patches, links and the observer. Turtles are agents that can move. The netlogo world is two-dimensional and is divided up into a grid of patches. Each patch is a square piece of ground over which turtles can move. Links are agents that connect two turtles. Links can be directed (from one turtle to another turtle) or undirected (one turtle with another turtle). There is only one observer and it does not have a location. Netlogo allow the user to play the system and exploring the behaviour based on some conditions [?]. The road of traffic is made by nodes and links. There have directed and undirected links which create the network of traffic system. Then cyclist behaves as agents and interact with the system. The real scenario of a bicycle system is turned into a virtual figure. From which we can observe realistic phenomena.

Bicycle simulation is popular recent research topic. Here exists several related works. The agent based simulation of bicycle makes a different substitute through appraise and compare by simulate a bike sharing system [?]. A research focus on augmentation of a tramway into a rail station square. The transportation service accessibility is increased through conjunction of public transportation services [?]. Another bicycle sharing simulation is done about to measure the accident probability of sulzberg city [Wallentin 2015].

2.6 Spatio temporal analysis

Spatio temporal analysis is an analysis where it is done on time and space related data. Both time and space information are managed by it. Large spatio-temporal database can be maintained and analyzed by it [Visser-Quinn 2019].

When data are accumulated with respect to time and space and having spatial and temporal property then the analysis type of any system is called spatio temporal analysis. It explains the amazement of an event's spatio temporal perspective at a particular time and location.

In bicycle model the analysis is done through cycle's riding time and its trip space. The cycles are used for trip in the specified area. When cyclists make their trip the position of bicycles is changed with respect to time. This phenomena creates a time-space relational data of the bicycle riding system. Now if we use the data for analysis then it is called spatio temporal analysis.

Spatiotemporal data analysis is an emerging research area due to the development and application of novel computational techniques allowing for the analysis of large spatiotemporal databases. Spatiotemporal models arise when data are collected across time as well as space and has at least one spatial and one temporal property. An event in a spatiotemporal dataset describes a spatial and temporal phenomenon that exists at a certain time t and location x. An example would be that of the patterns of female breast cancer mortality in the US between 1990-2010, where the spatial property is the location and geometry of the object – US states with breast cancer mortality rate information, and the temporal property is the timestamp or time interval for which the spatial object is valid – 1990-2010 breast cancer mortality years. Other applications for spatiotemporal analysis include cases in the domains of biology, ecology, meteorology, medicine, transportation and forestry.

The analysis of spatiotemporal data requires that both temporal correlations and spatial correlations be taken into account. Assessing both the temporal and spatial dimensions of data adds significant complexity to the data analysis process for two major reasons:

1. Continuous and discrete changes of spatial and non-spatial properties of spatio-temporal objects.

2. the influence of collocated neighboring spatio-temporal objects on one another.

2.6.1 Spatio-temporal Data Analysis Workflow

With some of these challenges in mind, we provide a walkthrough on how spatiotemporal analysis is done, using a largely generalized approach.

The goal is to give you enough information to know if you want to begin working with spatio-temporal data, how you could start to assess whether your data is appropriate, and give you resources to further the analysis.

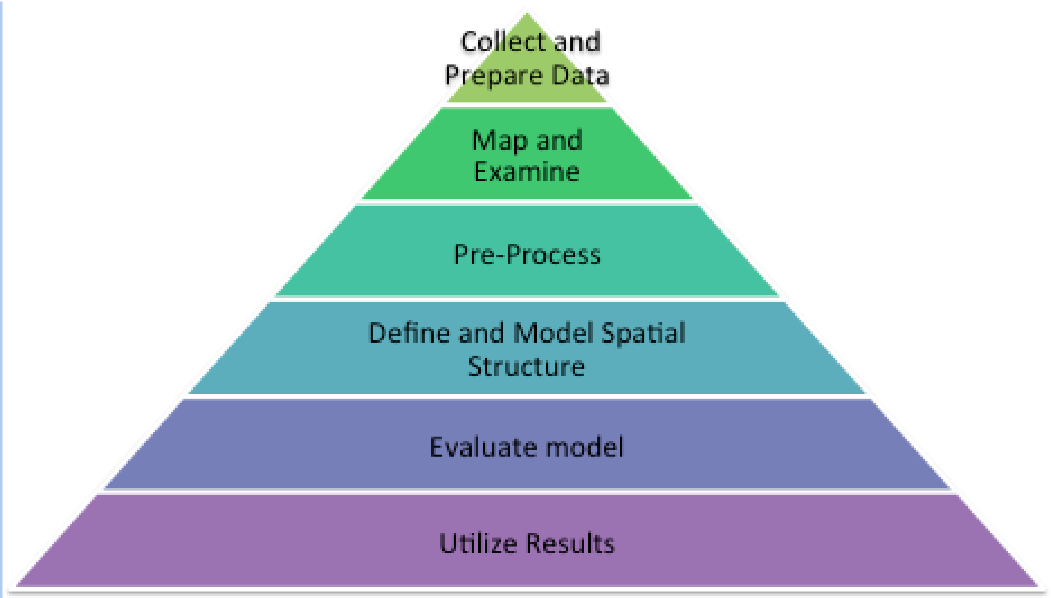

Figure 2.1: Spatio-temporal workflow.

In practice, the two primary goals of spatio-temporal analysis tend to be prediction and description. For simplicity, we will focus on the workflow that would occur as you were doing a descriptive spatio-temporal analysis with a chronic disease focus. However, this approach can branch into a predictive model, for which we provide resources to explore as need.

1. **Collect and Prepare Data :**

   The key requirement is that all data must be linked to both a spatial and a temporal component. Spatial data can analyze on many levels, zip codes, census tract, state, geocode, etc. Temporal data is often analyzed as multiple data points per observation over time and can be measured by just as many ways as the spatial data, if not more. One can also look at events by year, month, minute, second, but this must also be linked to spatial data.

   The plethora of options for how to define your spatial and temporal data is a challenge, particularly when you want to compare your results to other studies or make it applicable to other populations.

If you are collecting your own data, doing research before data collection is important to see how others are defining the problem. If you are fortunate enough to have primary data that you collected, you can define your observations with your analysis in mind. But often, this is not the case and we are analyzing data that we are obtaining from other sources. It is getting more frequent to be able to get information linked to spatial data, which really helps as you are collecting data for your analysis. When doing complex analyses like spatio-temporal analysis you need so many data components and can rarely find all data from only one source. Often you can collect some of the data yourself with your particular analyses in mind and then you can obtain other components from different secondary data sources. More and more databases are including spatial data leading to more and more people with the ability to do these analyses.

2. **Map and Examine :**

   Once we have our data, as in all data-driven studies, a key next step is to begin to examine the data. Much like when we run simple frequencies or cross-tabs to explore other forms of data, we do the same general approach with spatio-temporal data. In addition to descriptive data analyses, we can examine our data using simple descriptive maps. By doing this, we can get a clear visualization of important characteristics or trends that may be linked to spatial data that we may not see by just looking at the data. We can also pinpoint outliers, potentially erroneous data, and small or large cell counts that may become problematic.

3. **Pre-Process**

   Spatiotemporal data may often need to be transformed before analyzing. If necessary, use techniques to center the data and use transformations to make the data fit closely to a normal distribution. Another key aspect is to test for non-independence of spatially linked observations. Need to be concerned about clustering, and depending on what your data looks like and what clustering you are expecting, you use different methods. Various ways clustering can occur are :

   - Spatial clustering based on non-spatial attribute values of ST objects.
   - Clustering of moving objects.
   - Density clustering.

   If clustering is found you may need to transform data using algorithms which extract potential statistical clusters. Often in spatio-temporal data an issue that may lead to bias is the existence of auto-correlation. This stems back to the requirement we discussed earlier of the analytical models that all spatial objects are independent of each other and all temporal data is independent of it.

4. **Define and Model Spatial Structure :**

   There are many models that are housed within the spatio-temporal framework and that can be used for these types of analyses. We found that saying 'spatio-temporal analysis' was almost as broad as saying 'regression' which makes it simultaneously easy to analyze your data in these methods because you can fit many models and it is difficult to analyze because there is rarely a clear cut method to use.

5. **Evaluate Model :**

   To evaluate the quality of the model, the analyst then examines the model residuals.

   The temporal distribution of the residuals is explored by means of the time graph display and the spatial distribution by means of the map display. A model is considered correctly generated or captures the general features of spatiotemporal variation when there is an absence of clear temporal and spatial patterns, or in other words, the distributions for each dimension appear as random noise.

   If random distribution is not established the analyst may choose to modify the model or segment the group and revise the analysis. Other key factors to consider in the evaluation step is to look at the key assumptions of the theoretical ST model – all temporal structures should be captured by the smooth temporal basis function and the spatial dependencies should demonstrate stationarity.

6. **Utilize Results :**

   The last task in analytic approach with spatiotemporal data analysis it to utilize the results.

   Once the model has been satisfactorily built, adjusted and output checked, the results can be used in risk analyses and decision-making. Interpretation of the results depends on whether the model is built to describe novel patterns in health mapping or whether the model was developed to predict future disease outcome patterns.

Chapter 3

Description Of The Proposed Model

3.1 Statement of the model

Agent based modeling relate the agent and environment. Agent senses the environment and create interfaces on environment through actuator. We focus our attention on understanding the patterns and behaviours of bicyclists as a form of active transport. There are a few number of data sources which can be used to analyze patterns of cycling across coverage area. In contemporary society, importance of social research is increasing for business, public services and other activities. Many social phenomena have a spatio-temporal dimension and involve dynamic decisions made by individuals.

The jobike bicycle model reflects our current understanding of cycle traffic in Jahangirnagar University campus. Many types of ecological, environmental data are collected over discrete spatial and temporal domains. Input parameters were thus exclusively based on available statistical data, findings from literature and expert inputs. The parameters of interest comprised different, pre-defined types of cycling agents, their relative share of total cyclists and their respective characteristics [Wallentin 2015]. The scenario was designed to test the hypothesis on the importance of Jahangirnagar University's surrounding area for the spatial distribution of cyclists. This paper proposes an agent-based simulation system for spatio-temporal analysis of jobike movement. It can be used to visualize statistical characteristics of sampling method, effects of biases involved by behavior of respondents and design of survey considering such effects.

In Our model, jobike are used by customers. There are three types of customers in our bicycle model, student cyclist, working cyclist and guest cyclist. These three types of agents have different impact on this model. The day that was used for validation was selected randomly based on the validation day represents the spring season of the year, where most cyclists are longed, it further avoids the vacation at university to properly account for student cyclists, the weather was suitable for cycling. These types of simulations have allowed agents to move on some form of grid, most typically, as part of a haphazard network on an artificial landscape. Customer share ride when jobike is free. The availability of jobike is the main theme of the agent based system. At the end of servicing, the jobike are being congested in unorganized way at specific place. Specific term is used for busiest place. End of the day the agent collect these jobike for well distribution for the next service, so that these are available to customer. Our spatio-temporal analysis gives a result of the prediction about that where the maximum jobike is ultimately exist at the end of the day.

3.1.1 Model Architecture

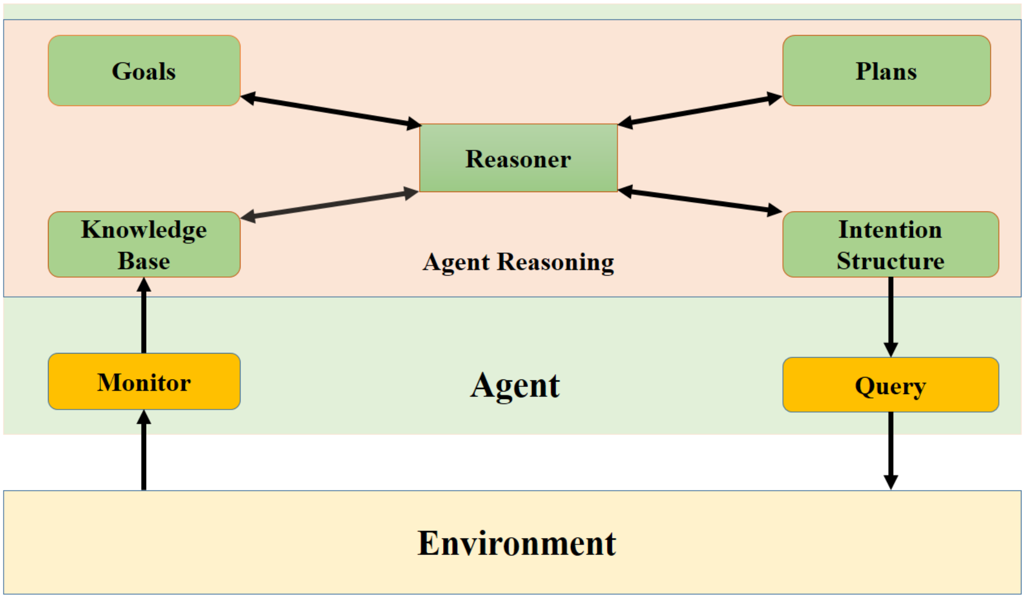

Figure 3.1: Architecture of Agent based model.

- **Agent :**

  A rational agent could be anything which makes decisions, like a person, firm, machine, or software. It carries out an action with the best outcome after considering past and current percepts in artificial intelligence, an intelligent agent refers to an autonomous entity which acts, directing its activity towards achieving goals, upon an environment using observation through sensors and consequent actuators Intelligent agents may also learn or use knowledge to achieve their goals.

- **Goals :**

  A goal is an idea of the future or desired result that a person or a group of people envisions, plans and commits to achieve. People endeavor to reach goals within a finite time by setting deadlines. Agents attain goals by adapting the behaviour of it.

- **Plans :**

  Agents are entities that act upon the world. Rational agents are those that do so in an intelligent fashion. What is essential to such an agent is the ability to select and perform actions. Actions are selected by planning, and performing such actions is a matter of plan execution. So the essence of a rational agent is the ability to make and execute plans. Rational agents embedded in a realistically complicated world may devote more time to epistemic cognition than to practical cognition, but even for such agents, epistemic cognition is in an important sense subservient to practical cognition [Pollock 1999].

- **Knowledge base**

  A knowledge base (KB) is a technology used to store complex structured and unstructured information used by a computer system. The initial use of the term was in connection with expert systems which were the first knowledge-based systems. A knowledge-based system consists of a knowledge-base that represents facts about the world and an inference engine that can reason about those facts and use rules and other forms of logic to deduce new facts or highlight inconsistencies.

- **Reasoner**

  In information technology a reasoning system is a software system that generates conclusions from available knowledge using logical techniques such as deduction and induction. Reasoning systems play an important role in the implementation of artificial intelligence and knowledge-based systems By the everyday usage definition of the phrase, all computer systems are reasoning systems in that they all automate some type of logic or decision. Reasoning systems come in two modes: interactive and batch processing. Interactive systems interface with the user to ask clarifying questions or otherwise allow the user to guide the reasoning process.

- **Intention Structure**

  This is introduces a new theory of intention representation which is based on a structure called a Dynamic Intention Structure (DIS). The theory of DISs was motivated by the problem of how to properly represent incompletely specified intentions and their evolution. Since the plans and intentions of collaborating agents are most often elaborated incrementally and jointly, elaboration processes naturally involve agreements among agents on the identity of appropriate agents, objects and properties that figure into their joint plans. [Hunsberger 2008].

- **Environment**

  An environment is everything in the world which surrounds the agent, but it is not a part of an agent itself. An environment can be described as a situation in which an agent is present. The environment is where agent lives, operate and provide the agent with something to sense and act upon it. An environment is mostly said to be non-feministic.

3.2 Purpose

The intention of the model was to simulate the spatio-temporal distribution of cyclists in the JU campus for proper or effective distribution of cycle. Here effective distribution reflects that cyclist attains cycle if they need. For this reason cycles are assigned in predefined station. The model was thus meant to be used as a base model to propagate and test hypotheses on driving factors of cycle traffic and their implications for traffic management and route preferences. From this result of simulation it becomes clear that approximation about bicycle distribution in specific location.

3.3 Agents of the model

Agents have internal states. These internal states can be represented by discrete or continuous variables. The agent's behaviour can be represented as a state-determined automata or finite state machine, a state transition occurs whenever the agent interacts with another agent [Arifin 2016]. Agents in this jobike bicycle model were individual cyclists who travel on the street network of JU campus. They use the bicycle when they need. Basically agents mean that who sence the envoironment and work on it through actuator. There were three different agent types: ‘working cyclists’, ‘student cyclists’ and ‘guest cyclists’(Fig. 2). The types of agents are defined according to their usage pattern of bicycle in this model. Each agent was assigned various trips: it commutes from any station to its respective destination and back again. Regardless their type, agents cycled at an average speed of 12km/h. The system state of interest was the spatio-temporal ordination of cyclists. The temporal extent of a simulation was one weekday from morning to night with a resolution of one minute.

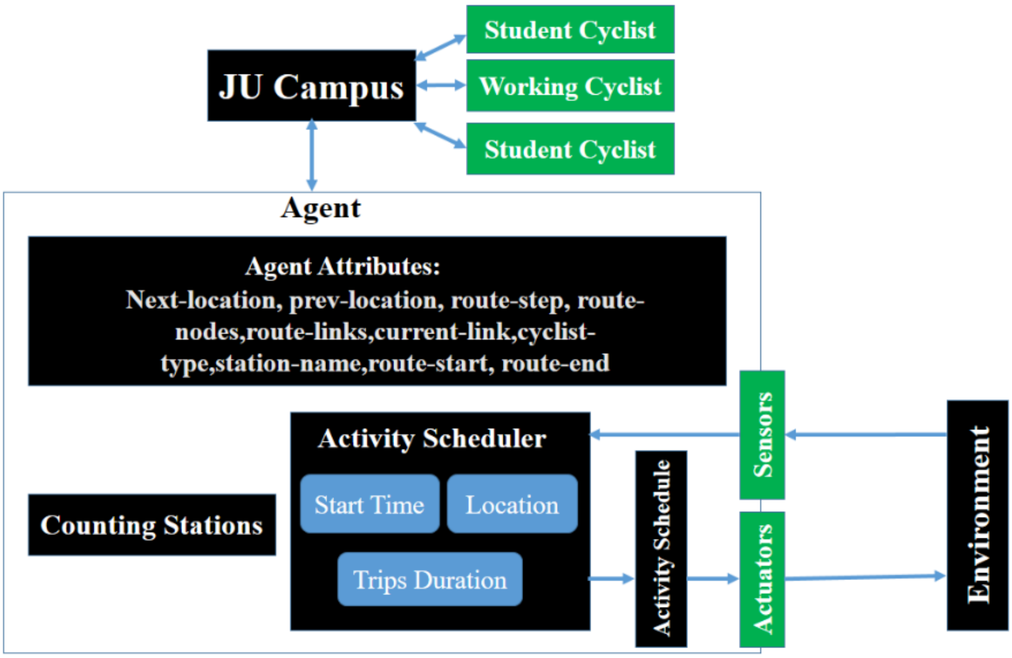

Figure 3.2: Simulated bicycle agent architecture.

Chapter 4

Implementation Of The Proposed Model

4.1 Implementation of the model

For bicycle traffic simulation Netlogo which is an agent-based adaptation model and also an traffic assignment model. The agents of the model have more plans which is a chain of activities. That includes agents location and time. Agent's activities are connected through links. Here link means route of transport.

The recurrent open circuit of netlogo consists of some components, such as The transport network loading, executing of the selected plans in a artificial reality. Next, all executed plans are scored, by a active method, which is based on its genuine representation. Ultimately all specified agents are granted to re-plan. They use their memory for generating new plan. Such as agents can use new route for transporting system. The adaptation activity is continuing until the system is determined as a relax system [Ziemke 2017].

The netlogo network comprised of links and nodes. The coordinate are stored as nodes. The links are used to make a connection with these nodes. Different types of nodes are used as start node, end node and intermediate nodes. The links has link length, receptivity of flow and storage etc.

Geographical information system (GIS) is the typical main data source for Netlogo, especially for the establishment of networks and initialize the netlogo world. GIS also constructs the data source of this study. Additional attributes used to describe properties of the infrastructure, which are relevant for the decisions of cyclist.

Several procedures are used to carry out this model: such as setup, go, reset, add-stations and add-cyclists. Where "setup" method, initializing the world street address by loading street dataset and initialized different types of cyclist's random movement. Another method "go" explains tick procedures and shows the movement of cyclists. The method named "reset" is used to clear all of the properties of previous netlogo world. The function "add-stations" adds all of the counting stations in the world. The method "add-cyclist" creates different types of cyclist and targets on random vertices and setting the location, target and schedule for different types of cyclists. Also it visualizes the cyclist and the target, compute the shortest-path between cyclist and target.

4.1.1 Coding

Here we provide specific source code of some procedures.

```
;; create street network
to setup
  clear-all
  reset-ticks


    set status "Loading street dataset ..."
    set edges-dataset gis:load-dataset "netlogodata/edges.shp"


  gis:set-world-envelope gis:envelope-of edges-dataset

  let x-ratio (item 1 gis:envelope-of edges-dataset - item 0 gis:envelope-of edges-dataset) / ( max-pxcor - min-pxcor )
  let y-ratio (item 3 gis:envelope-of edges-dataset - item 2 gis:envelope-of edges-dataset) / ( max-pycor - min-pycor )

  ;; the greater ratio defines the correct scale factor between GIS coords and NetLogo coords
  ifelse x-ratio > y-ratio [set scale-factor x-ratio][set scale-factor y-ratio]
  ;; convert km/h from user input to m/min and further convert meter to NetLogo units; 1 tick thus is 1 min
  set avg-speed (avg-cycle-speed / scale-factor) * 300 / 3.6

  ; ignore push regulations
  set avg-push-speed avg-speed ;/ 5

  ;; load employees raster
  set status "Loading employee dataset ..."
  set employees-dataset gis:load-dataset "netlogodata/Employees_250m_raster.shp"
  gis:apply-coverage employees-dataset "BESCH01" employees
  set max-employees max [ employees ] of patches

  set status "Loading resident dataset ..."
  set residents-dataset gis:load-dataset "netlogodata/guest_250m_raster.shp"
  gis:apply-coverage residents-dataset "HWS01" residents
  set max-residents max [ residents ] of patches

  set status "Loading students dataset ..."
  set students-dataset gis:load-dataset "netlogodata/student_250m_raster.shp"
  gis:apply-coverage students-dataset "SUM_STUDEN" students
  set max-students max [ students ] of patches

  set status "Loading counting-stations dataset ..."
```

```
set stations-dataset gis:load-dataset "netlogodata/counting-stations.shp"

if write-file? = true [prepare-file-output]

ask patches [
  ifelse employees < 0 OR employees >= 0 [][set employees 0 ]
  ifelse residents < 0 OR residents >= 0 [][set residents 0 ]
]

let street-segments gis:feature-list-of edges-dataset
let total-segments length street-segments
let processed-segments 0

;; loop line features (= street segments)

foreach street-segments [

  ifelse gis:property-value ? "oneway_FT" = 1 [ set oneway_FT? true ] [set oneway_FT? false]
  ifelse gis:property-value ? "oneway_TF" = 1 [ set oneway_TF? true ] [set oneway_TF? false]
  ifelse gis:property-value ? "restrictio" = 1 and gis:property-value ? "restrict_1" = 1 [ set street-push? true ] [set street-push? false]
  ifelse gis:property-value ? "restrictio" = 2 and gis:property-value ? "restrict_1" = 2 [ set street-restricted? true ] [set street-restricted? false]
  set street-index_FT gis:property-value ? "index_FT"
  set street-index_TF gis:property-value ? "index_TF"
  set street-length gis:property-value ? "Shape_Leng"
  let street-name gis:property-value ? "NAME1"

  ;; loop vertex lists for each street segment
  foreach gis:vertex-lists-of ? [
    let i 0 let len (length ?) - 1

    let first-vertex nobody
    let last-vertex nobody
    let current-vertex nobody
    let previous-vertex nobody
    let street-segment nobody
    let pathdirectional []
```

```
    foreach ? [
      let coords gis:location-of ?
      if not empty? coords [

        ;; if junctions-only = "on"
        ;;    only consider start and end points of a street segment (= nodes of the graph) for the network graph
        ;; if junctions-only = "off"
        ;;    consider all vertices of a street segment for the network graph
        if i = 0 or i = len or not junctions-only [
           create-vertices 1 [
              set xcor item 0 coords
              set ycor item 1 coords
              set hidden? true
              set pathends []
              set paths-nodes []
              set paths-links []

              set current-vertex self
              ;; use existing vertex at this location if there is one and mark new vertex for deletion
              let existing-vertex one-of other nav-vertices-here with [ xcor = [xcor] of myself and ycor = [ycor] of myself ]
              if existing-vertex != nobody [
                set current-vertex existing-vertex
                set die? true
              ]

              ;; create link to previous vertex and save the street type characteristics
              if previous-vertex != nobody and previous-vertex != current-vertex [
                   ask current-vertex [
                     ifelse oneway_FT? = false and oneway_TF? = false
                     [ create-street-with previous-vertex [ set street-segment self] ]
                     [
                       if oneway_FT? = true [ create-oneway-from previous-vertex [ set street-segment self ] ]
                       if oneway_TF? = true [ create-oneway-to previous-vertex [ set street-segment self ] ]
                     ]
                     if is-link? street-segment [
                       ask street-segment [
                         set push? street-push?
                         set restricted? street-restricted?
                         set name street-name
```

```
                        set segment-length link-length * scale-factor
                        ;; for simplicity reasons, the index is only one-directional
                        set weight (segment-length * (street-index_FT ))]
                    ]
                  ]
              ]

              set previous-vertex current-vertex

              ;; add this vertex to the a temporary list (of all vertices of this polyline)
              set pathdirectional lput current-vertex pathdirectional

              if i = 0 [ set first-vertex current-vertex set breed nav-vertices]
              if i = len [ set last-vertex current-vertex set breed nav-vertices]

              if die? = true [ die ]
           ]
        ]
      ]
      set i i + 1
    ]

    ;; create auxiliary direct link between origin and destination for shortest path computation
    ask first-vertex  [
      if self != last-vertex [

         ; create regular, bidirectional street
         if oneway_FT? = false and oneway_TF? = false and street-index_FT = street-index_TF
         [ create-nav-street-with last-vertex [set street-segment self]
           ; set routing weight
           if is-link? street-segment [
             ; ((1 + Indexwert ) * 5 -4 ) * Shape Length
             ask street-segment [ set push? street-push? set restricted? street-restricted? set segment-length street-length set weight (segment-length * ((1 + street-index_FT ) * 5 - 4)) ]
           ]
         ]

         ; create a street with 2 lanes that have direction-dependent routing indices
         if oneway_FT? = false and oneway_TF? = false and street-index_FT != street-index_TF



    [ create-nav-oneway-from last-vertex [set street-segment self ]
      ; set routing weight
      if is-link? street-segment [
        ask street-segment [ set push? street-push? set restricted? street-restricted? set segment-length street-length set weight (segment-length * ((1 + street-index_FT ) * 5 - 4)) ]
      ]

      create-nav-oneway-to last-vertex [set street-segment self ]
      ; set routing weight
      if is-link? street-segment [
        ask street-segment [ set push? street-push? set restricted? street-restricted? set segment-length street-length set weight (segment-length * ((1 + street-index_TF ) * 5 - 4)) ]
      ]
    ]

    ; create a oneway from to
    if oneway_FT? = true
    [ create-nav-oneway-from last-vertex [set street-segment self ]
      ; set routing weight
      if is-link? street-segment [
        ask street-segment [ set push? street-push? set restricted? street-restricted? set segment-length street-length set weight (segment-length * ((1 + street-index_FT ) * 5 - 4)) ]
      ]
    ]

    ; create a oneway to from
    if oneway_TF? = true
    [ create-nav-oneway-to last-vertex [set street-segment self set hidden? true ]
      ; set routing weight
      if is-link? street-segment [
        ask street-segment [ set push? street-push? set restricted? street-restricted? set segment-length street-length set weight (segment-length * ((1 + street-index_TF ) * 5 - 4)) ]
      ]
    ]
  ]
]

;; hide navigation links
ask nav-streets [ set hidden? true ]
ask nav-oneways [ set hidden? true ]
```

```
 ;; compute paths of each vertex to the start and end vertices of the path that it belongs to
     foreach pathdirectional [ask ? [
         let p-nodes []
         let p-links []
         if last-vertex != ? and position last-vertex pathends = false [
           set pathends lput last-vertex pathends
           set p-nodes sublist pathdirectional position ? pathdirectional length pathdirectional
           set paths-nodes lput p-nodes paths-nodes
           set p-links get-node-links p-nodes
           set paths-links lput p-links paths-links
         ]
         if first-vertex != ? and position first-vertex pathends = false [
           set pathends lput first-vertex pathends
           set p-nodes reverse sublist pathdirectional 0 (position ? pathdirectional + 1)
           set paths-nodes lput p-nodes paths-nodes
           set p-links get-node-links p-nodes
           set paths-links lput p-links paths-links
         ]
     ]]
   ]
   set processed-segments processed-segments + 1
   set status (word "Setup Streetnetwork " round (processed-segments / total-segments * 100) "%")
  ]
  ;; visualize "special"streets
  ask oneways [
    set shape "oneway"
  ]
  ask (link-set streets oneways) [
    if push? = true [ set shape "push" set thickness 0.5]
    if restricted? = true [ set shape "restricted" set thickness 2]
  ]
  ;; add counting-stations
  add-stations
end
```

4.1.2 Initialization and Input Data

The agent-based bicycle model encounters some points of location that works as a stationary points. Traffic model that has been successfully applied to the Jahangirnagar University campus. Users of bicycle are randomly generated in the different zones of the service region according to a Poisson distribution with mean equal to the average trip generation rate in the zone for the current time step [Vogel 2014]. Here each cyclist generates the starting location by random selection from a holding density grid. A simulation means basically animating a model with respect to a real phenomena and transform it to model through computation and visualization.

Agent-based model's main research inspiration is visualization. In this model, an auxiliary output window features a clock, calendar and display the month, day, year, 24-hour time, the day of the week (Sunday-Saturday). This clock is used to track and record interactions and spread of the jobike movement. Six primary location schematics are modeled for agents to interact within: hall, department, two gate and other workspaces. A transport network is created to simulate the customer to and from different places while arriving at different times during the day.

There are three types of cyclist such as "working","student" and "guest". Soucre and destination of cyclists are differed among cyclist types. Working cyclists have a destination, which is generated randomly from a workplace density grid. Student cyclists start their trip from various university locations. Guest cyclists select destinations preferably within campus areas . There are approximately 15000 students in campus and about 53% (8000 out of 15000) use jobike. There are nearly 1400 working people and about 42% (600 out of 1400) use jobike. Approximately 2000 guest in campus and about 10% (200out of 2000) use jobike.

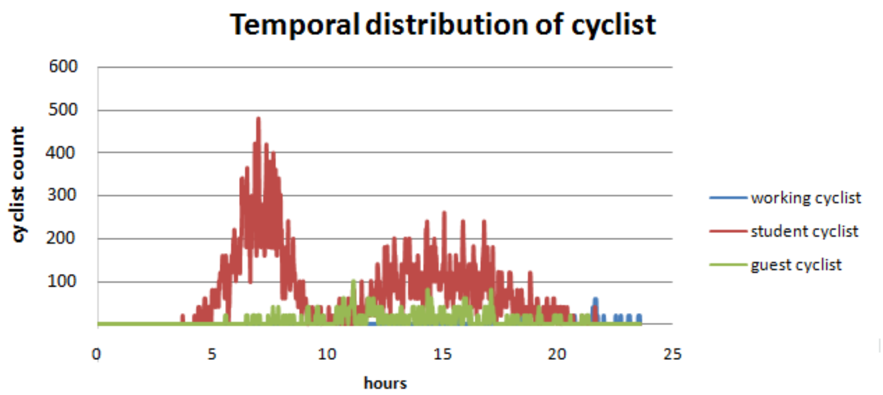

Figure 4.1: Cycling agent's temporal distribution in normal day.

Starting and returning times were chosen randomly by each individual cyclist, following normal distributions (Fig. 4.1), working cyclists start in the morning around 10 am and return home after about eight hours work. Students cycled to university around 9.00am and they may go to other stations such as social science, bottola, dairy, prantik, new arts and old arts (Fig. 4.2). Trips to guest activities were scheduled to start either at 10.00am and they visit various places in campus.

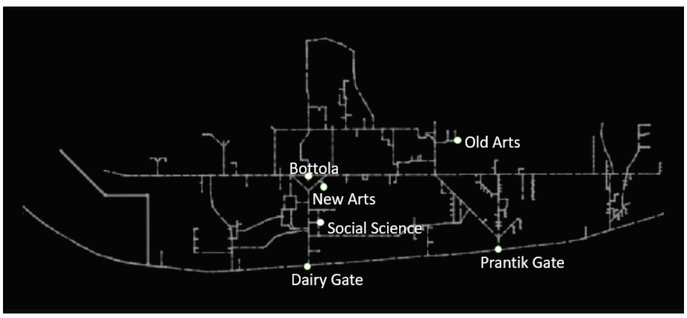

Figure 4.2: Counting stations's location.

In holiday, starting and returning times were chosen randomly by each individual cyclist, following normal distributions (Fig. 4.3). The number of guest cyclist is increased.

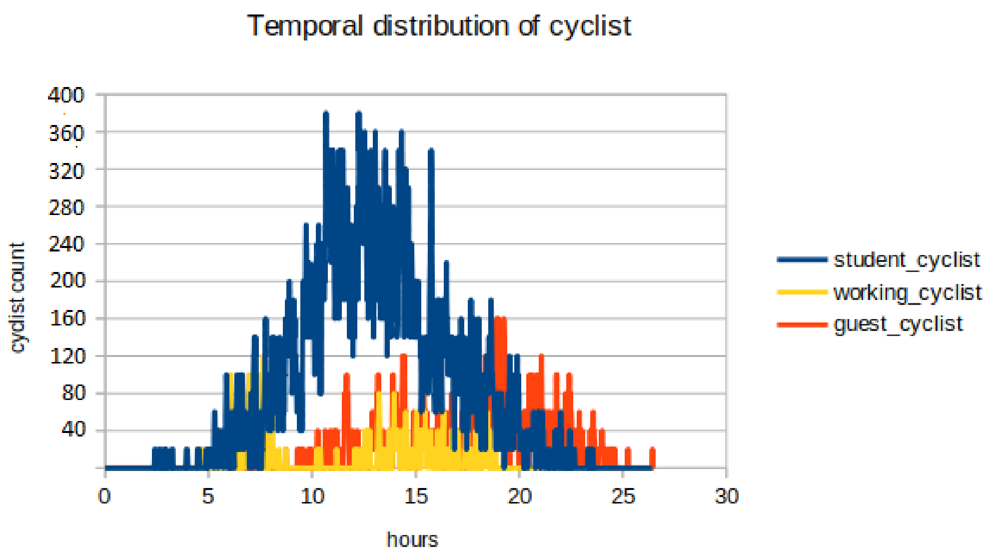

Figure 4.3: Cycling agent's temporal distribution in holiday.

In the NetLogo software street networks were then rearranged to a routable network dataset. Routes were calculated using the Dijkstra algorithm of NetLogo's network extension weighted by a street-segment safety index.

Chapter 5

Result and Validation

5.1 Result

Agent based modeling focuses on the individual active element of a system. This is modeled in dynamic way which means an abstract system. Processes are applied on discrete components [Huynh 2015]. In jobike movement simulation observing the spatial distribution of jobike means distribution according to places. Here the model is simulated through ten runs and taking the average of following runs. Spatial and Temporal distribution of jobike movement with trips's heatmap is given below.

5.1.1 Simulated bicycle trips's heatmap

A heatmap is a two-dimensional representation of data in which values are represented by colors. A simple heatmap provides an immediate visual summary of information. More elaborate heat maps allow the viewer to understand complex data sets. There can be many ways to display heat maps, but they all share one thing in common, they use color to communicate relationships between data values that would be would be much harder to understand if presented numerically in a spreadsheet.

From (fig. 5.1) heatmap that the visualization of jobike model at JU campus. The heatmap showed that the road is highlighted from dairy to bottola via social science and new arts and the road of cycle is also highlighted towards the prantik gate. So that most of the jobike is appeared in those places.

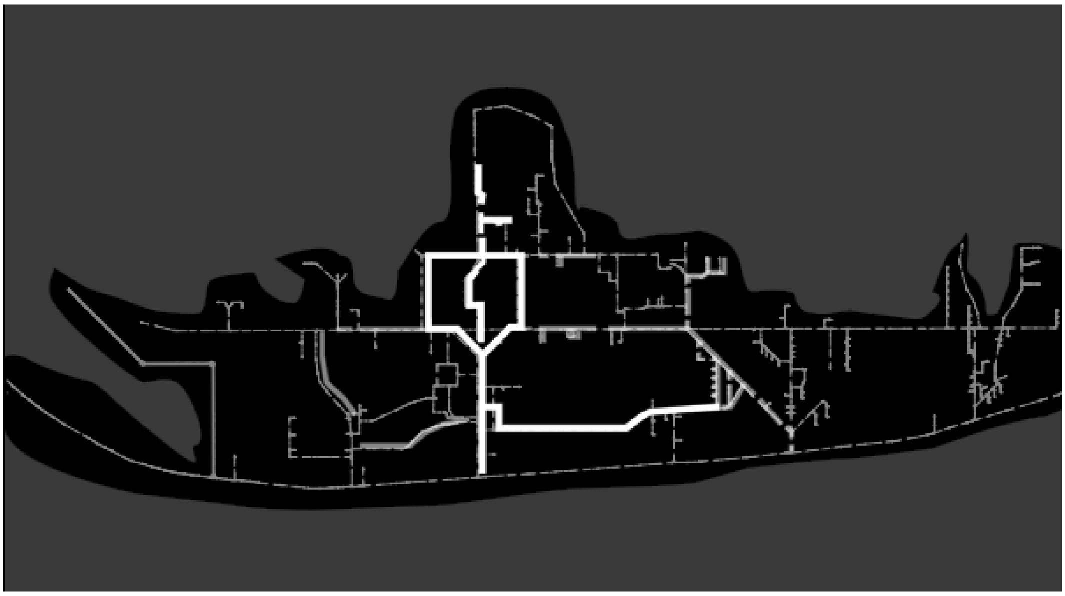

Figure 5.1: Simulated bicycle trips's heatmap. The most crowded areas of Jahangirnagar University are highlighted.

5.1.2 Simulated result for normal day

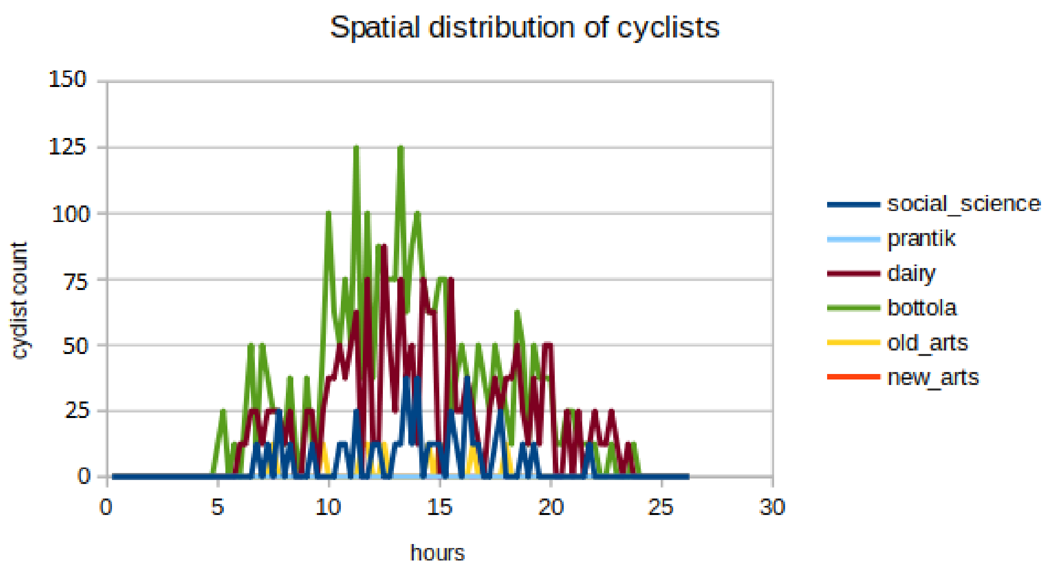

Figure 5.2: Cycling agent's spatial distribution in normal day.

In this model, here different perspectives of jobike distribution are emerged. (Fig. 5.2) shows when ju campus is in its regular or normal day, the place named bottola is more crowded with people. The cycles are seen most in bottola, dairy gate station is also crowded but less than bottola and the other places like social science, new arts, old arts and prantik are less crowded so the cycles are less seen there. Again the number of cycles is more than holiday.

5.1.3 Simulated result for holiday

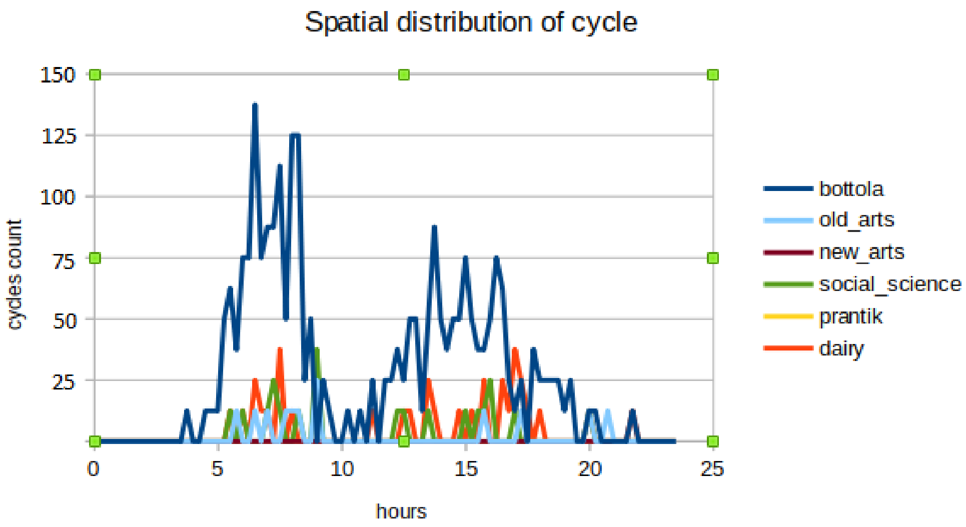

Figure 5.3: Cycling agent's spatial distribution in holiday.

But the scenario is different in holiday in ju campus. In holiday the timing of bike sharing is different from the normal days. The distribution of cycles among the cyclists are slightly changed. The number of cycle is less than the normal day and also the number of guest cyclist is increased. From the (Fig. 5.3) we see that the graph cyclist count vs time, during 9.00am to 10.00am and 1.00pm to 3.00pm most of the cycles are appeared in dairy gate to new arts region. During 1.00pm to 3.30pm most of the cycles are appeared in bottola. The temporal distribution of simulated bicycle trips correctly showed two maxima, one in the morning and one in the afternoon. In the morning, cycle traffic starts with an sudden increase of rides and slowly fall in the evening.

5.2 Validation

The results of this research show that bicycle flow patterns on the scale level of the entire city of Jahangirnagar University are truly emergent phenomena. As in any complex system, higher level flows spatio-temporal patterns could not be intuitively anticipated from the conceptual model.

In order for any model to produce reliable and useful results, it needs to be valid enough. In fact, the validity is often considered the most important property of the models [Junges 2012]. A process of quantifying the validity by determining whether the model is an accurate representation of the studied system is called validation and needs to be done thoroughly and throughout all the phases of model development [Law 2009].

The simulation of agent-based bicycle model is matched with physical movement of jobike. The distribution of cycle in specified area of JU campus is as like as the ordination of simulated work.

Chapter 6

Conclusion

6.1 Conclusion

This research accomplished in a bottom up approach in dynamic system for a hypothetical result of jobike distribution. Spatio-temporal analysis of jobike movement is efficient for good servicing system. It helps to predict the availability of bicycles through simulation in agent-based modeling. Developing bicycling through skillful design result in diminishing hardship. For attaining these motive, a greater perception of the spatial impression on bicycle distribution is required. The simulation results in a scenario that the jobikes are used by different types of agents in different times. At the end of the day jobikes are collectively seen such selective places. That is the point of our research. The outcome of our research is properly adjusted to physical movement of jobike bicycle traffic system.

Bibliography

[Abar 2017] Sameera Abar, Georgios K Theodoropoulos, Pierre Lemarinier et Gregory MP O'Hare. *Agent based modelling and simulation tools: a review of the state-of-art software*. Computer Science Review, vol. 24, pages 13–33, 2017.

[Arifin 2016] SM Niaz Arifin, Gregory R Madey et Frank H Collins. Spatial agent-based simulation modeling in public health: design, implementation, and applications for malaria epidemiology. John Wiley & Sons, 2016.

[badu osei] Frank badu osei. *Improving Public Health through Geographic Information.*

[Cascetta 2001] Ennio Cascetta et Maria Nadia Postorino. *Fixed point approaches to the estimation of O/D matrices using traffic counts on congested networks*. Transportation science, vol. 35, no. 2, pages 134–147, 2001.

[Hunsberger 2008] Luke Hunsberger et Charles L Ortiz. *Dynamic intention structures I: a theory of intention representation*. Autonomous Agents and Multi-Agent Systems, vol. 16, no. 3, pages 298–326, 2008.

[Huynh 2015] Nam Huynh, Vu Lam Cao, Rohan Wickramasuriya, Matthew Berryman, Pascal Perez et Johan Barthelemy. *An agent based model for the simulation of transport demand and land use*. UCL STEaPP, 2015.

[Junges 2012] Robert Junges et Franziska Klügl. *How to design agent-based simulation models using agent learning*. In Proceedings of the 2012 Winter simulation conference (WSC), pages 1–10. IEEE, 2012.

[Kaziyeva 2018] Dana Kaziyeva, Gudrun Wallentin, Martin Loidl, Stefan Mohr et Christian Neuwirth. *Reviewing Software for Agent-based Bicycle Flow Models*. GI_Forum 2018,, vol. 1, pages 291–296, 2018.

[Law 2009] AM Law. *December. How to build valid and credible simulation models*. In Simulation Conference (WSC), Proceedings of the, pages 24–33, 2009.

[Newell 1972] Allen Newell, Herbert Alexander Simon*et al.* Human problem solving, volume 104. Prentice-Hall Englewood Cliffs, NJ, 1972.

[Nikolai 2009] Cynthia Nikolai et Gregory Madey. *Tools of the Trade: A Survey of Various Agent Based Modeling Platforms.* Journal of Artificial Societies and Social Simulation, vol. 12, no. 2, page 2, 2009.

[Pollock 1999] John L Pollock. *Planning agents.* In Foundations of rational agency, pages 53–79. Springer, 1999.

[Rostow 1960] Walt W Rostow. The stages of growth: A non-communist manifesto. Cambridge University Press Cambridge, 1960.

[sandtable 2017] sandtable. *why-agent-based-modelling.* 2017.

[Singh 2009] Vivek Kumar Singh et Ashok K Gupta. *Agent based models of social systems and collective intelligence.* In 2009 International Conference on Intelligent Agent & Multi-Agent Systems, pages 1–7. IEEE, 2009.

[Singh 2011] Vivek Kumar Singh, Swati Basak et Neelam Modanwal. *Agent Based Modeling of Individual Voting Preferences with Social Influence.* In Dhinaharan Nagamalai, Eric Renault et Murugan Dhanuskodi, editeurs, Trends in Computer Science, Engineering and Information Technology, pages 542–552, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Smith 2007] Eliot R Smith et Frederica R Conrey. *Agent-based modeling: A new approach for theory building in social psychology.* Personality and social psychology review, vol. 11, no. 1, pages 87–104, 2007.

[Vandenbulcke 2011] Grégory Vandenbulcke, Claire Dujardin, Isabelle Thomas, Bas de Geus, Bart Degraeuwe, Romain Meeusen et Luc Int Panis. *Cycle commuting in Belgium: Spatial determinants and 're-cycling'strategies.* Transportation Research Part A: Policy and Practice, vol. 45, no. 2, pages 118–137, 2011.

[Visser-Quinn 2019] Annie Visser-Quinn, Lindsay Beevers, Lila Collet, Guiseppe Formetta, Katie Smith, Niko Wanders, Stephan Thober, Ming Pan et Rohini Kumar. *Spatio-temporal analysis of compound hydro-hazard extremes across the UK.* Advances in Water Resources, 2019.

[Vogel 2014] P Vogel, Jan F Ehmkeb et Dirk C Mattfelda. *Decision support for tactical resource allocation in bike sharing systems.* Technische Unveristaet Braunschweig, 2014.

[Wallentin 2015] Gudrun Wallentin et Martin Loidl. *Agent-based bicycle traffic model for Salzburg City.* GI_Forum J. Geogr. Inf. Sci, vol. 2015, pages 558–566, 2015.

[Wilensky 1999a] U Wilensky. *Netlogo.[online] Netlogo 6.0. 1.* Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Available at:< http://ccl. northwestern. edu/netlogo/>[Accessed 1 Mar. 2017], 1999.

[Wilensky 1999b] Uri Wilensky *et al.* *Center for connected learning and computer-based modeling.* In NetLogo. Northwestern University, 1999.

[Wilkinson 2018] David Wilkinson. *What is Agent-Based Modelling.* 2018.

[Ziemke 2017] Dominik Ziemke, Simon Metzler et Kai Nagel. *Modeling bicycle traffic in an agent-based transport simulation.* Procedia Computer Science, vol. 109, pages 923–928, 2017.