高度道路交通システムにおける情報伝送、ユーザ体験および持続性の研究

メタデータ	言語: eng
	出版者:
	公開日: 2019-06-25
	キーワード (Ja):
	キーワード (En):
	作成者: チョウ, ジョウ ホウ
	メールアドレス:
	所属:
URL	https://doi.org/10.15118/00009911

Transmission Performance, User Experience, System Sustainability in Urban Intelligent Transportation System

Chaofeng Zhang

Department of Information and Electronic Engineering
Muroran Institute of Technology

This dissertation is submitted for the degree of
Doctor of Philosophy of Computer Science and Engineering

March 2019

Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

Chaofeng Zhang
March 2019

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Mianxiong Dong and Prof. Kaoru Ota for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof.Jay Kishigami, and Prof. Kazuhiko Sato, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

My sincere thanks also goes to Prof. Ota Kaoru, and Prof. He Li, who provides very valuable advice for my daily research as well as spiritual encouragement. Without they precious support it would not be possible to conduct this research.

I would like to acknowledge the financial, academic and technical support of the Muroran Institute of Technology, Japan. The Department of Information and Electronic Engineering has provided the support and equipment I have needed to produce and complete my thesis. I would also like to acknowledge the financial support and warm regards from China Scholarship Council as well as Embassy of China, Japan, which has been with me for three years.

Last but not the least, I would like to thank the lab members: Mr. Liangzhi Li, Miss. Wuyuzhaola, Mr. Jianwen Xu and Miss. Caijuan Chen, for technical support and spiritual encouragement throughout this research and my life in general. I hope all the team members can get the degree they desired.

Chaofeng Zhang
March 2019

Abstract

ITS (Intelligent transportation system) is a modern way of urban transportation management. By improving road throughput, managing traffic, and enhancing human mobility, it can improve the service performance of urban transport networks in a holistic way. Ultimately, through better traffic management, congestion can be slowed and traffic safety improved. The large-scale deployment of ITS is very meaningful, which can greatly affect all aspects of urban life. The traditional means of traffic management can not deal with real time traffic management very well. The advanced ITS, in keeping with the times, is the integration of as much IoT as possible (Internet of Things) equipment, for the human to provide new service experience.

Recently, the spectrum band beyond 60GHz has attracted attentions with the growth of traffic demand. Previous studies assumed that these bands are not suitable for vehicle communications due to the short range and high rate of blockage. However, it also means there is no existing service or regulation designed for these bands, which makes this area free to apply. One purpose of this study is to construct an applicable map of 0.1-1 THz supported autonomous vehicle system through the study of channel capacity, autonomous relaying and self-driving. For sending a large volume of data, the vehicles keep a gigabit level link through line of sight channels, which share information about traffic condition and captured videos for further processing. The full use of this research can significantly improve the performance of autonomous driving. A brief overview of possible waveforms followed by specific channel characteristics in 0.1-1 THz band is discussed and an autonomous relay system for the gigabit road communication is presented. At last, we describe how the high-rate short-range communication helps complement extra data to support the advanced new self-driving technologies.

To improve the user experience, taxi sharing service aims to share the taxi resource efficiently between passengers with specific relevant plans. However, due to the lack of social interactions, it is hard to improve user experience without considering passengers' inner connections. To overcome this problem, we propose a cloud-based taxi-sharing system, which provides real-time taxi sharing services both considering the practical travel needs and QoS in the journey. The purpose of this service is to minimize the travel cost through sharing

calculated schemes reasonably and quickly and improve the sharing experience as much as possible. We model the citywide level map into blocks and indexed, in order to search for the potential paths of ride-sharing dynamically and intelligently. Meaning while, we explore the paths among communities to find suitable companions as well. Finally, synthesizing the suggestions of edge exploration on road network and social network, we design a heuristic algorithm called SONETS to achieve the designing goal and make a real deployment. As well as the establishment of the mobile application and online service, we also conduct a large-scale simulation to evaluate the performance with others. The result shows that the proposed solution could make the whole taxi-sharing system less calculation complexity, higher sharing QoS, and most of all, keeping the minimum cost of travel cost within 1.16% increment than distance optimal algorithms.

To keep the sustaibility of urban ITS, we also investigate the maintenance of city-wide sharing pedelecs, which apply battery-powered motor to assist pedaling and significantly extend the riding coverage. The large scale deployment of pedelecs, however, requires a careful design of maintainance system to replace the batteries of pedelecs regularly, which can be very costly. This study develops an optimal path planning scheme for replenishment trucks based on big data analytic and prediction. To the best of our knowledge, we are the first to use the systematic methodology of Analysis/Modeling/Test to advance the bike prediction process using the captured datasets, and conduct the hybrid prediction process learns internal connections between different dataset dimension as well as optimize the overall maintenance efficiency.

Table of contents

List of figures

List of abbreviations

Acronyms / Abbreviations

2D Two-Dimensional space

3C Computing, Communication and Control

3D Three-Dimensional space

ACO Ant Colony Optimization

ANN Artificial Neural Networks

ATLR Autonomous Relay Algorithm

BSRC Bicycling Sharing Radar Cross Model

CO Cost-Optimized Algorithm

CPS Cyber-Physical System

DSRC Dedicated Short Range Communications

DTN Delay-tolerant networking

GA Genetic Algorithm

GBit Gigabit

GPS Global Positioning System

HD High-Definition

IFFT Inverse Fast Fourier Transform

IMOA Intelligence Sharing Bike Maintenance Optimization Algorithm

IoV Internet of Vehicles

IR Impulse Radio

ITS Intelligent Transportation System

LiDAR Light Detection and Ranging

LOS Line of Sight

MAC Media Access Control

mmWave millimetre Wave

NLOS No Line of Sight

NYC The City of New York

PC Personal Computer

Pedelec Pedal electric cycle

POMDPs Partially Observale Markov Decision Processes

PSBL Intelligent Transportation System

QoE Quality of Experience

QoS Quality of Service

RCP Raised-Cosine Pulse

SONETS Social Network based Heuristic Algorithm

Thz Terahertz

TSP Traveling Salesman Problem

UAV Unmanned Aerial Vehicle

V2V Vehicle to Vehicle

Chapter 1

Introduction

1.1 Background and Motivation

ITS (Intelligent transportation system) is a modern way of urban management, mainly contains advanced information technology, communication technology, sensing technology, control technology and computer technology. By improving road throughput, managing traffic, and enhancing human mobility, it can improve the service performance of urban transport networks in a holistic way. Ultimately, through better traffic management, congestion can be slowed and traffic safety improved. The new generation of mobile devices also brings a better way of user interaction, from the user's point of view to enhance the overall service experience.

Its large-scale application is very meaningful, and it can greatly affect all aspects of urban life. First, it can reduce the growing traffic jams. Traffic jams can cause some problems, including environmental pollution,Increased energy consumption and the feasibility of a scheduled plan. Second, it can reduce the probability of traffic accidents by driving assistance systems. Because about three-fourths of traffic accidents are caused by human error. Finally, it reduces the transport burden on each line, thus maximizing the overall throughput of the transport network

The traditional means of traffic management can not deal with the above problems very well. Simple restrictions on travel will bring great inconvenience to people, and building new roads is time-consuming and laborious. And some traffic aids, such as cameras, GPS (Global Positioning System) and microwave detectors, they play a very limited role.

ITS, in keeping with the times, is the integration of as much IoT as possible (Internet of Things) equipment, for the human to provide new service experience. For example, by the analysis and induction of vehicle trajectory, we can optimize the driving path. Through the tracking, identification and density analysis of vehicles, we can analyze and extract special road conditions. The addition of artificial intelligence provides new technical support for automatic driving and traffic signal recognition.

1.1.1 Transmission Performance of Spectrum Availability in Urban ITS

The fast growing demand for vehicle communications has raised the importance of spectrum availability, which is used for data creation, sharing, and consumption. With this trend, each mobile cell has to keep 10-GBit transmission rate around the year of 2020 [31]. Some already well designed and researched communication systems applied in 60GHz may be theoretically compatible with more than 0.1-THz bandwidth. However, it is difficult to establish a stable and efficient link due to the greater attenuation in vehicle communications. A couple of research groups all around the world have started to investigate the abundant spectrum resource to be operated beyond 300-GHz, the so-called THz communication systems [11]. With the advanced physical layer solution realized in the future, the new spectral bands can be utilized on Internet of Vehicles (IoV) [21], which lays the foundation for the challenges due to the explosive growth of autonomous vehicles' data transmission, sharing and consumption.

1.1.2 User Experience using Taxi Sharing in Urban ITS

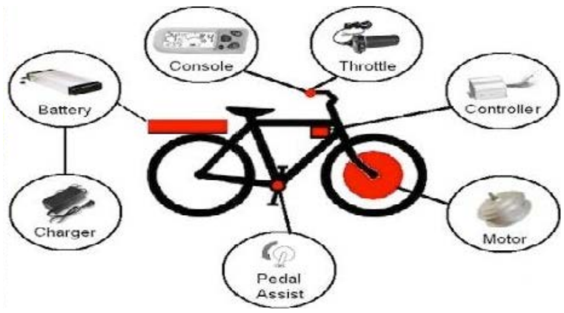
New styles of public resource management emerge due to the development of smart devices, which significantly enhance the quality of citizens' life. Online taxi sharing is one potential solution for public transportation, which aims to solve the problem of limited taxi resource in citywide. The main advantage of this service is that it relieves the pressure on the transportation and improves the quality of user experience [59]. In an online taxi sharing service, the system provides more sharing opportunities and gives out better routes through the interactions of the smart device's interface. Since the service is provided in real time and the sharing is also a green transportation style, this taxi sharing management problem has already generated much interesting in both academy and industry in recent years [2].

Some general studies have already put forward achievements in this service: mobile-cloud architecture based taxi-sharing system makes it feasible to satisfy the request of consumers with minimum increase of travel distance, where a dynamic spatiotemporal index based taxi routing algorithm is used to find the solution[43] [16], where exists a coordination problem between car capacity, precedence and user constraints. Generally, traffic congestion and environmental concerns are the main advantages that inspire consumers to use taxi sharing services [8]. However, some potential problems still prevent people from experiencing the services if we ignore passengers' special needs [54]. For example, many countries have to set priority seat and the non-smoking area in urban transportation to fulfill passengers' particular needs; The difference in the opposite sex and the strangeness of passengers make the oppressive atmosphere much like a bus ride. Moreover, if considering the coordination between consumers, the differences of departure time and destination are the reasons why most of the sharing agreements cannot be satisfied. The waste of waiting time and the division of trip costs make the sharing arrangement hard to establish.

1.1.3 System sustainability about E-bike sharing in urban ITS

Bike-share has recently become a rising trend worldwide as the last-mile solution in cities from the bus/subway station to home[53]. In a bike-sharing system, riders can use their smartphones to locate and use a shared bike available in the nearby. When finish riding, riders can either return the bikes to a bike station or merely leave bikes on the roadside until the next rider.

(a) Pedelec is stored at the roadside stations



(b) One pedelec contains the components of the battery, console, controller, and motor

(c) The maintenance truck stops at the roadside to provide battery exchange service

(d) The installation and removal of the battery are less than 10 seconds

Fig. 1.1 Pedelec's share system, structure and battery maintenance

An issue of the legacy bikes is that people can quickly get tired, especially in windy weather or on steep roads, making the service coverage of sharing bikes limited to about 2km. Pedelec is a new type of e-bike to solve this issue. The pedelec applies a battery-powered motor to assist the pedaling as in Fig. 1.1 and accordingly saves the energy of riders. When the pedalling speed is above a reasonable speed (e.g., 25 kilometres per hour), indicating a

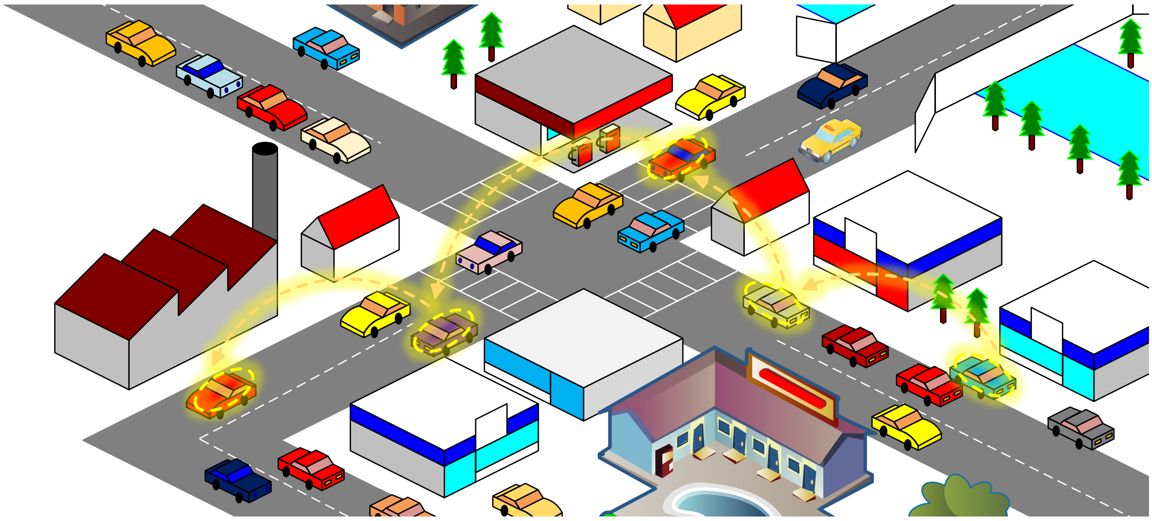


Fig. 1.2 No line of sight data sharing through terahertz vehicle networks

smooth and pleasant ride, the pedelec bike cuts off the electric power to save its energy. As a result, pedelec can extend the ride range to 5km and completely replace vehicles as the last mile solution.

1.2 Purpose of This Study

The purpose of this study is to improve the transmission performance, user experience and system sustainability in urban intelligent transportation systems.

To improve the transmission performance, we construct an applicable map of 0.1-1 THz supported autonomous vehicle system through the study of channel capacity, autonomous relaying and self-driving. Fig.1.2 is an example of data sharing scenario through terahertz band. For sending a large volume of data, the vehicles keep a gigabit level link through line of sight channels, which share information about traffic condition and captured videos for further processing. If we can overcome the fading issues, the full use of this research can significantly improve the performance of autonomous driving. At first, the motivations from the existing technologies applied in autonomous vehicles need be fully discussed and then a brief overview of possible waveforms should be investigated followed by specific channel characteristics in 0.1-1 THz band. Discussions about the platform and throughput in developing technology and demonstrator are also given. An autonomous relay system for the gigabit road communication will be presented next and we also need to describe how the high-rate short-range communication helps complement extra data to support the advanced new self-driving technologies, where all the above problems should be fully discussed.

To improve the user experience, we develop a new social network based taxi sharing system. Although some constraints dismiss consumers' idea of sharing, people still frequently share their trip with whom are in their social connections [13]. As social network is partly based on real-world relationships (family, colleague), passengers are more likely to trust the companions if the relationship is based on the social network rather than some purely online services (second life, online games, etc.) [6]. The connections among them are more reliable than complete strangers. Therefore, in this thesis, we analyze to discover one of the various properties in social networks, the community structure. When looking for suitable travel companions on the map, the social community structure could provide suggestions about potential passengers in close communities. The proposed method to build the social communities is clustering. However, the difference with previous studies is that the connections of vertexes in one community are dense, while the interactions between communities are much sparse [46][6]. In Fig.1.3, it shows an example of a social network community structure after the analysis. This partitioning method for social networks has efficient meaning to detect the communities. For instances, the communities being mined may share the same social groups in the real world, or some common interests and backgrounds; Research areas often separate communities in schools, and in e-mail based communication networks are often separated by different companies; Since taxi sharing is an opportunity that all passengers sit in a small space, the companions in the same or close communities are more reliable than strangers. It is more likely to improve the interactions between each other, and most of all, to make the social connections tiger, which could naturally improve the QoS (quality of services) during the journey as well as other intangible benefits [29]. In a word, identifying the communities in local social networks can help the system discovery and understand the properties of passengers, and further improve the ride-sharing experience through throughout arrangement.

To improve the system sustainability, we investigate the E-bike sharing system called pedelec sharing. However, a key challenge of pedelecs is how to manage their batteries. In specific, a fully charged battery in the pedelec provides power within 8 hours [38], and therefore, one pedelec may need to have its battery replaced for three times a day when continually used to provide 24-hour service [44]. To keep the service of pedelecs, replenishment trucks are driven across the bike sharing stations to change the batteries of pedelec. Note that the efficiency of replenishment is not only key to the service quality of the entire pedelec system, but also determines the maintenance cost. This gives rise to the path planning issue of replenishment trucks. The issue is a challenging task because that with different locations and surrounding environments of bike sharing stations, pedelecs stored

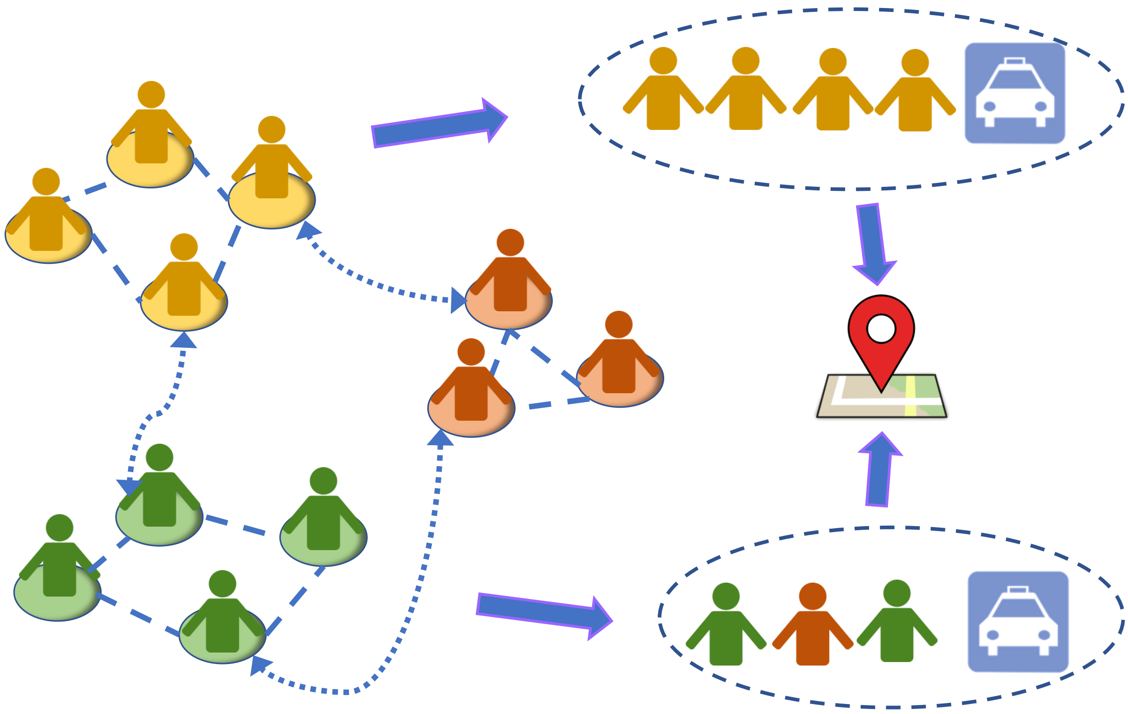


Fig. 1.3 Social network community structure applied in taxi sharing service

have very different battery usage, and within one station, pedelecs's battery usage can also be quite heterogeneous due to the wide service coverage and different riding history.

1.3 Originalities

To apply Thz transmssion in V2V (vehicle to vehicle) network, previous studies assumed that these bands are not suitable for vehicle communications due to the short range and high rate of blockage. However, it also means there is no existing service or regulation designed for these bands, which makes this area free to apply. Therefore, in this study, we draw a potential map of terahertz vehicle transmission for autonomous vehicles to break the blockage of short range and unstable links. Firstly, we give a brief overview of possible waveforms followed by the specific channel at 0.1-1 THz. Then, we propose an autonomous relay algorithm called ATLR for the gigabit level communication in the high-speed road environment. Finally, we discuss how the gigabit links help relieve interference problem and provide extra data to support various instructions in autonomous vehicles.

To apply social network into taxi sharing, the service we developed aims to share the taxi resource efficiently between passengers with specific relevant plans. However, due to the lack of social interactions, it is hard to improve user experience without considering passengers' inner connections. In this study, we introduce the social network community detection method into the allocation of online taxi sharing service. We propose an optimization scheme integrates social interaction to improve the sharing efficiency and user experience, by handling the trade-off between cost and social interaction. Then, we state and solve the sharing arrangement problem by computing a three-step algorithm called SONETS which includes community detection, bidirectional taxi searching, and social network based heuristic algorithm, in order to satisfy overwhelming requests with limited taxi resource in rush hour. A real-world based mobile application and simulation are conducted, using local social network accounts. The results show that our service can fit well with the economic demand and improve the QoS (quality of service), to achieve cost efficiency, fairness, and humanized designing.

To discuss the system suitability, at last, we investigate the maintenance of city-wide sharing pedelecs, which apply battery-powered motor to assist pedaling and significantly extend the riding coverage. The large scale deployment of pedelecs, however, requires a careful design of maintainance system to replace the batteries of pedelecs regularly, which can be very costly. This study investigates the issue by developing an offline solution in two steps. First, an optimal and efficient hybrid predictive prediction model, combining the future usage increment from local predictor, global predictor and inflection predictor, is developed to improve the prediction accuracy. The developed approach can predict the usage demand in every 48 hours on a scale of millions of pedelecs. Second, based on the result of big data analytics developed, an optimal path planning scheme for the replenishment trucks is developed. As compared to other schemes, our IMOA can save 40% of cost. To verify our proposal, extensive real-data driven simulations are performed which show that the accuracy of the prediction process is high enough then each traditional method and IMOA solves the maintenance problem efficiently.

1.4 Main Contribution

The main contribution are discussed as following:

1. **Impulse waveforms** are discussed in this paper due to the fine multipath and fading environments. It enables precise ranging and data transmission using IEEE 802.15.4a. These features are indispensable for autonomous system, which makes the motivation for us.

2. **Relay algorithm** helps to improve the quality of real-time multi-hop links, by using embedded sensors to monitor the whole candidate space. It handles the channel

condition and road presentation accurately, and leverages the relative position efficiently. After the position adjustment of autonomous vehicles, the terahertz channel can approach optimized performance.

3. **Road simulation** is conducted, with the scenario that our sports car cannot detect the truck which is changing lane in the front. A decision of left-handed rotation from the autonomous driving system may put our car into a dilemma due to the lack of road information.

4. **Social network** relationship is introduced into the taxi resource management when limited taxi resource meets with the rush hour. The community structure of a social network is taken into consideration when scheduling socially connected passengers in one taxi.

5. **Optimization problem** is formulated and calculated using our proposed 3-step process, including community detection, real map taxi searching and social network based heuristic algorithm called SONETS. It searches for an optimized solution to make a cost efficiency, fairness, and humanized service designing.

6. **Smart device application** called 'Social Network Taxi Sharing' is designed and applied in the real environment. Furthermore, the large-scale simulation is deployed based in citywide, using real user information and cartographic information.

7. **Big data based prediction** is used to develop an efficient offline prediction scheme based on big data analytics. We use local predictor to analyze the characteristics of the station itself, global predictor to analyze global correlations, and inflection predictor to deal with unexpected situations. An prediction integrator is used at last to improve the prediction accuracy.

8. **Optimal path planning algorithm** is developed here. Based on the predicted results, we can analyse the degree of correlation between stations in real time. After making a station correlation map, a new pedelec maintenance algorithm is developed.

9. **Evaluation and verification** is conducted here. We select NYC's shared bicycle data and city-relevant big data for real simulation experiments. After comparison of prediction accuracy, real-time average power and overall service quantity, it is proved that our solution is superior to most benchmark algorithms.

1.5 Organization

This dissertation is organized in the following: Chapter 2 discusses the fundamental problems and studies about ITS, including communication issues, cost issues and efficiency issues. Chapter 3 discusses how to solve the performance issue in urban ITS, where the next-generation vehicle networking technology using Thz is conducted here. Chapter 4 proposes a real time and real environment solution to meet with the experience issue in urban car sharing services, where the user-oriented taxi sharing service using social network is considered in this study. Chapter 5 introduces the efficiency-oriented maintenance service using AI prediction, to show a comprehensive solution how to deal with the sustainability issues with the development of urban ITS. Finally, Chapter 6 is related to conclusion and future work.

Chapter 2

Fundamentals and Related Work

This chapter describes some fundamentals and related works that will be used in this study. The chapter introduces the fundamental studies about the urban ITS system, concluding terahertz supported self-driving system, urban taxi sharing system and green ITS transportation, respectively.

2.1 Terahertz Band for Self-Driving ITS

2.1.1 Terahertz Communication Model

Terahertz band communication is envisioned as a key wireless technology to satisfy the demand of data collection in autonomous vehicles, by alleviating the spectrum scarcity and capacity limitations of widely used 4G or future 5G networks. Meanwhile, it motivates the potential of existing applications in vehicle technologies. The THz band is generally considered as the spectral band that spans the frequencies between 0.1 and 10 THz. Frequency regions below and above this band have been fully investigated: Spectral band below 0.1 THz (known as the microwave) is not available to support gigabit level links [65] while the spectral band beyond 10 THz has too many constraints to realize a feasible optical approach for mobile wireless communications. Therefore, the THz band is still one of the least explored frequency bands for communication. There are many challenges in the realization of efficient and practical THz band communication networks, which is related to the development of innovative solutions in the different layers. The solutions overcome the challenges in mm-Wave systems can also be utilized in terahertz band, which bring enough bandwidth for gigabit level transmission. Impulse waveforms are discussed in this paper due to the fine multipath and fading environments. It enables precise ranging and data transmission using

IEEE 802.15.4a. These features are indispensable for autonomous system, which makes the motivation for us.

2.1.2 Terahertz Autonomous Relay

Autonomous relaying technologies can be discussed to solve the capacity and feasibility problem mentioned above, but there are still several disadvantages for the real application. First, the range of terahertz band is relatively shorter than existing wireless technologies such as Wi-Fi and 4G technologies. The attenuation of transmission distance is much higher than longer waves and the antenna array is also not completely developed. Next, the obstacles can easily block terahertz waves, which is similar with mm-Wave. Furthermore, the molecular absorption such as water vapor molecules can affect the channel performance more significantly [3]. To overcome the disadvantage mentioned above, vehicle relays can be applied to form Ad hoc networks or delay tolerant networks (DTN) among autonomous vehicles. These networks can bypass the obstacles such as barriers or constructions and extend the communication links to other vehicles which are out of the shorter terahertz communication range. The autonomous vehicles are more suitable than man-driven vehicles since the minor position adjustment can be made to achieve better multi-hop performance. Here, we propose an autonomous terahertz relay (ATLR) algorithm to determine the relaying position. This algorithm helps to improve the quality of real-time multi-hop links, by using embedded sensors to monitor the whole candidate space. It handles the channel condition and road presentation accurately, and leverages the relative position efficiently. After the position adjustment of autonomous vehicles, the terahertz channel can approach optimized performance.

2.1.3 Terahertz Autonomous Driving

Autonomous driving in up-to-date studies mainly focuses on 'correctness', such as assuming driving operations are prescribed in advance as functions of time or state of the system, which can never be proven that autonomous vehicles are superior to human drivers [36]. Although predictive controls such as decision trees, partially observable Markov decision processes (POMDPs), and methods based on multi-policy decision-making are fully discussed, the lack of fully collected data prevents these control systems from the real deployment. For example, Line of sight (LOS) image data, self-positioning coordinates collected by one vehicle's sensors sometimes are inaccurate and needed further calibration. Instead, with the constant high-speed channels, the autonomous control system can make the decision depending on added invisible information, calibrated cooperative positioning and traffic condition of the

Fig. 2.1 Supplement of NLOS (no line of sight) traffic information for autonomous decision-making system

whole city. Fig.2.1 is an example of autonomous driving supported by additional traffic information. Because of the huge truck ahead, our sports car cannot detect the truck which is changing lane in the front. A decision of left-handed rotation from the autonomous driving system may put our car into a dilemma due to the lack of road information.

Nowadays, image recognition is important for human-like driving systems, but researchers only use the RGB data collected by individual sensors. Terahertz band can satisfy the requirement of constant high-rate video data transmission. This mean provides the dead zone image data or even computation capability through cloud service linked RSUs (Road-Side Unit) [32]. Traffic information is captured by the onboard sensors or other vehicles. With the added data provided by other vehicles, we conduct and discuss the impact of terahertz band for autonomous driving systems.

2.2 Taxi Sharing for Social Improvement

2.2.1 Urban Taxi Sharing

Taxi sharing comes into notice by researchers in the field of city resource management, especially for the environment concerning. Most of the studies are concentrated on how to save cost for passengers or improve the revenue for drivers, by endeavoring on the coordination of travel plans. For example, Huang analyzes the Shanghai and Hunan price linkage mechanisms for taxi services [26]. By advocating the efficient fuel usage of taxis, they find it useful for local governments to reduce the financial burden, which leads to the cleaner environment. They also consider one main problem in taxi sharing is how to build a reasonable schedule to save fare and time. Another innovative taxi sharing service in Chen's study [6] is that the system searches for passengers whose destinations are close enough in order to reduce the number of activated taxis. They implement a dynamic taxi-sharing system called Intelligent Transportation Systems (ITS), which decreases the total number of activated taxis efficiently and leads to the saving of fuel. Besides, Chen proposes a recommendation mechanism for taxi-sharing [10], which has the feature of lower taxi reaction time and cost. A non-cooperative game theory is used in that paper for selecting taxis in the same area, and finally, the solution is completed by the Nash equilibrium. Especially, considering both the profit of driver and quality of experience (QoE) for passengers, Zhang propose a QoS-aware taxi-sharing system with appealing the above two challenges [62]. By monitoring these requirements and dynamically adapting the schedules, they design a heuristic algorithm under acceptable delay, and it satisfies the needs of a large number of taxis. However, these works all assume passengers are willing to share all the time and have no extra demand for their companion, which is the lack of personalized customization. It may prevent the real deployment.

2.2.2 Social Network

Social Network relationship is introduced to improve the quality of taxi services. However, since social network is a weak relationship network, quantizing the abstract social network connection is hard. By researching different structure of social networks, Zhang concludes a novel model to describe the relationship strength between the different account of social network [61]. They find that it will be little impact on the information interaction if ignoring weak connections in social networks. On the opposite, the strong connections will create a strong influence on the dissemination of information, which brings more social interaction. Instead, if we could find communities on the map, the members sit together will be much

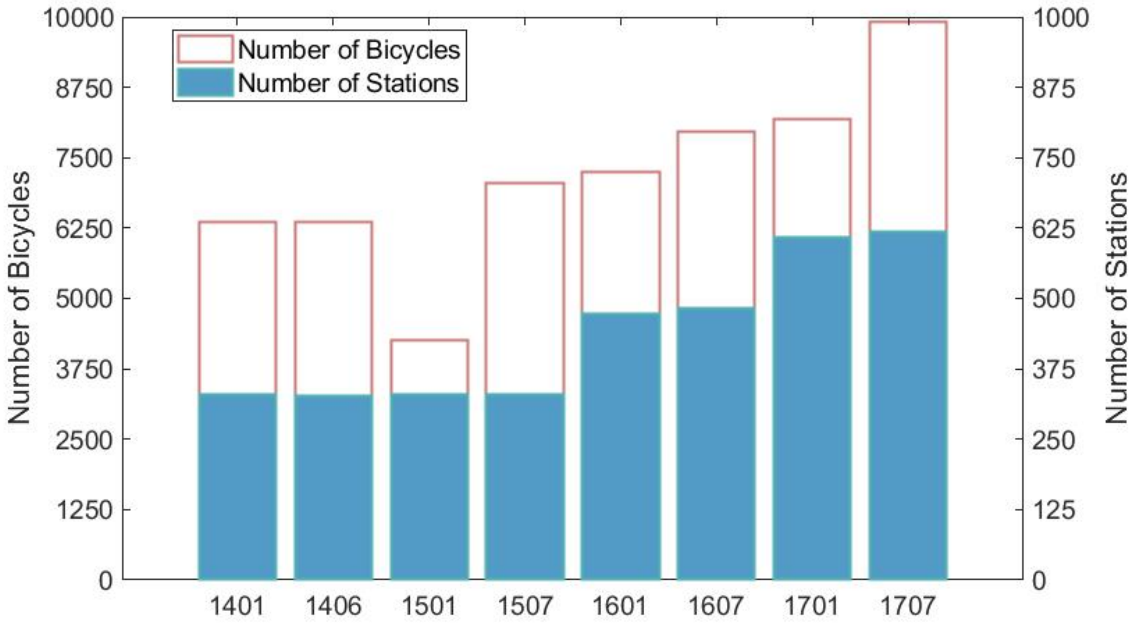


Fig. 2.2 Number of bikes and stations in NYC bike sharing is increasing year by year

easier to achieve the pleasant journey. Never than less, in real life, many social network communities are created by geographically closed people. For recommending social network friends who are geographically close, [60] propose a three-step statistical recommendation approach to create a GPS-enabled cyber-physical social network. Based on the experiment data of real-life data, they clear the advantage of merging GPS data and social network. The results show their novel method can bring extra social benefit in daily life using the GPS-based cyber-physical social network. Therefore, if we apply the social network flexibly in daily life, it could benefit the normal human social activities in an all-around way.

In a word, most of the above taxi sharing methods try to reduce travel cost and increase drives' benefit. However, none of them considers the passengers' inner needs. Finding an optimal tradeoff between cost and social benefit for passengers could increase the QoE significantly. Different from the works above, we start this taxi sharing problem by road network and social network processing, where we try to allocate passengers to improve the user experience when traveling.

2.3 E-bike Sharing for Green ITS transportation

There are two main problems should be solved during our study, one is the traffic flow prediction, and another is the bike maintenance service optimization. The traffic flow prediction should enormously combine the big citywide data as much as possible, to fit well with the characters of a city and improve the prediction accuracy. Then, based on the reliable predictive results, the system could analyze the inner connection between the outputs and make a more efficient pedelec service system.

2.3.1 Traffic Flow Prediction

Traffic Analyzing and Prediction solves the traffic problem from new sights. Pan [51] detects and describes the sudden change of traffic using human mobility and social media. Accidents, celebrations, disasters and the citywide events could result in a significant change of traffic flows, and they detect these unusual changes based on drivers' routing behaviors. They develop a system that identifies the anomalies of driver's routing behaviors based on their regular patterns and representative terms on social media. Hoang [23] proposes a new planning method to describe the traffic flows in different regions. They use the common citywide big data, including human mobility, weather reports, and road networks. three different categories of movement patterns are decomposed and analyzed by the system, including periodic patterns, changes, instantaneous changes. Then, they use Gaussian Markov Random Fields to exploit spatial-temporal dependence among different regions. Kong [33] propose a new estimation and prediction for traffic congestions using car trajectories. They use a fuzzy comprehensive evaluation method to leverage the real traffic flow patterns. An article swarm optimization algorithm is used to analyze this patterns and covert the inflection state to congestion situations, and their claim the proposed method is accuracy, instantaneity, and stability. These methods are highly dependent on real-time data, which is similar to this paper. However, their useful and customized predictions could not efficiently convert into ordinary citizens' cognitive services, which will be discussed deeper in this paper.

2.3.2 Traffic Maintenance Service Optimization

Maintenance Service Optimization is how to make the flexible use of sensed data and collected information to improve the QoE. Torres [53] aims to improve the sustainability and livability for the city and present a participatory sensing system called Becity to take the advantages of the collective knowledge of transportation cyclists, to improve the QoE of cycling. By combining the city usage information, cycling associations and government entities, the system provides bike routes with the shortest and accessible route, which is better than Google Maps. To maintenance the damaged bikes and reduce costs, Yao [58] proposes a hybrid bike allocation schedule to balance the system load effectively. The bike-sharing system in Washington D.C. is estimated, and their load-unbalance status could be reduced since the proposed system could significantly cut down the percentage of the bikes need to replace. Caggiani [5] discuss a multi-objective model to assist cyclists to keep lower travel cost, air pollution, and road safety. The new optimization model uses geolocated real-time data, which considering traffic congestions and flows, to select an optimal path and avoid air pollution or more safety. Then, they set up the graphics interface to guide users follow the current optimized path, and finally, result in the improvement of the cycling experience.

From these works we can find that, it is necessary to combine with real-time usage information and other geographical information together, to improve the efficiency and quality of urban traffic services. Instead of using recently uploaded information to 'adjust' the original schedule and 'adopt' new changes, our work aims to 'predict' the future trend and make a countermeasure at the beginning, which is more efficient and less risky comparing with the 'problems occur then to be fixed' model.

Chapter 3

Breaking the Blockage for Big Data Transmission: Gigabit Road Communication in Autonomous Vehicles

This chapter describes discusses how to solve the performance issue in urban ITS, where the next-generation vehicle networking technology using Thz is conducted here.

The purpose of this paper is to construct an applicable map of 0.1-1 THz supported autonomous vehicle system through the study of channel capacity, autonomous relaying and self-driving. For sending a large volume of data, the vehicles keep a gigabit level link through line of sight channels, which share information about traffic condition and captured videos for further processing. We believe the full use of this research can significantly improve the performance of autonomous driving. The remaining part of this paper is organized as follows:

In Chapter 3.1, we give a brief overview of possible waveforms followed by specific channel characteristics in 0.1-1 THz band. Discussions about the platform and throughput in developing technology and demonstrator are also given. An autonomous relay system for the gigabit road communication is presented in Chapter 3.2. In Chapter 3.3, we describe how the high-rate short-range communication helps complement extra data to support the advanced new self-driving technologies. Chapter 3.4 provides conclusions and an outlook to future research required in this area.

3.1 Communication Model

3.1.1 Impulse radio waveform

A few number of waveforms are considered for terahertz band communication as the key wireless technology to satisfy the constantly changing attenuation on the road. Impulse radio (IR) waveform is one of the suitable waveforms for vehicle communication, which takes the advantage of detecting traffic condition and certain resistance with NLOS transmission [12]. The short pulse can reduce the interference of other radio frequency systems, when the galloping vehicles have to cross by several static or mobile communication systems. In short range, the NLOS signal is obtained due to the relatively strong penetration of road obstacles. The distortion and spurious signal detection is also reduced due to the multipath propagation. In a word, the impulse radio waveform is suitable for terahertz band communications either from capacity aspect or attenuation aspect.

Generally, carrierless waveforms such as IFFT pulse are used to realize vehicle positioning technology or other traffic detecting technologies. Carrier waveforms such as raised-cosine pulse (RCP) shape are used for data transmission. The basic shape is $p(t) = h(t)cos(2\pi ft)$, when $h(t)$ could be Rectangular, Gaussian, Hann or Hamming pulse. In order to calculate the capacity of this channel model, we analyze the frequency domain for this pulse first. Fig. 3.1 is the PSD (Power Spectral Density) of Hann pulse waveform using four different settings. The frequency of the carrier wave f_c is set as 0.3, 0.6, and 1 THz. The PSD of this pulse is

$$W(f) = T_p[\frac{1}{1-(2fT_p)^2}]sinc2\pi T_p \tag{3.1}$$

where the duration T_p of pulse is 0.1 ns. From the figure we can observe that the available bandwidth increases with the frequency of carrier wave. It is noticed that with the increase of frequency, the width of the main lobe is also increasing. It broadens the analytical band and frequency resolution becomes worse. Therefore, Higher frequencies are suitable for less interference communication environments.

3.1.2 Channel Capacity

After the spectral analysis, we discuss the possible throughput of our proposed terahertz model. Generally, the total pass loss model $A(f,d)$ includes the sum of spreading loss and molecular absorption [28]. The noise includes system noise created by the electronic devices, antenna noise created by the Omni-antenna and molecular absorption noise depending on different humidity and air quality. For simply discussing the potential, we consider an

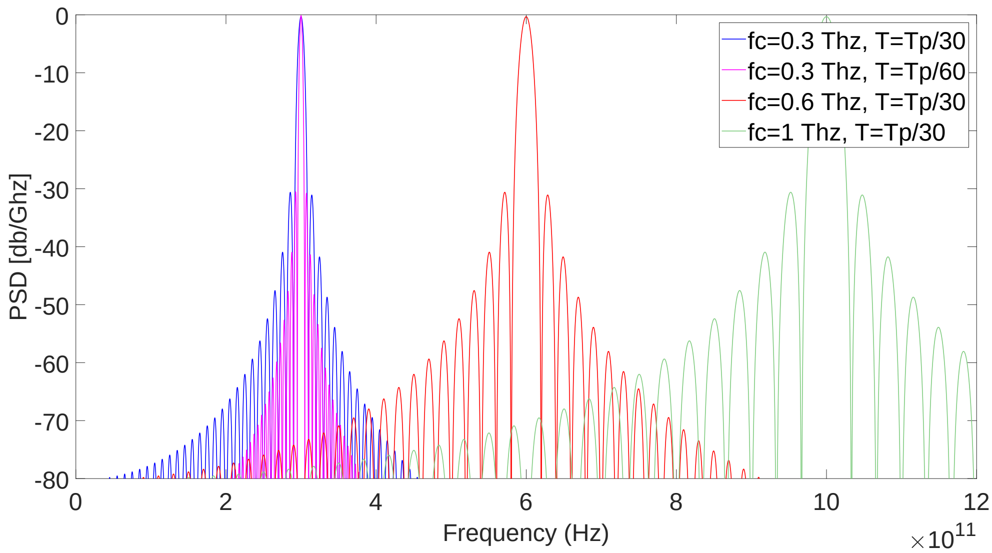


Fig. 3.1 The Hann pulse waveform for four values in THz Band

omni-antenna situation. Notice that the main challenge and decisive factor to improve the link performance is still the real deployment of directional antennas.

The capacity can be calculated by the sum of each sub-bands' capacity, noted as Δf. In each sub-band, we consider the PSD of each transmission wave and the noise are locally flat, therefore the channel capacity is

$$C(d) = \sum_i \Delta f log2[1 + \frac{S(f_i)}{A(f_i,d)N(f_i,d)}] \tag{3.2}$$

where d is the distance between Rx and Tx, S is the PSD function, A is the total pass loss function and N is the noise function. The capacity mainly depends on the distance and the SNR (Signal to Noise Ratio). The attenuation can be independently solved by relaying technology we introduced later. Meanwhile, the path loss and system noise increase with the frequency, and we can choose different pulses to achieve suitable PSD to improve the capacity.

In Fig.3.2, it shows the capacity as a function decreases with the distance. Generally, the molecular absorption noise becomes extremely high at several specific narrow bandwidth [3], which divides the whole THz band into 4 windows (0.38-0.44, 0.45-0.52, 0.62-0.72, 0.77-0.92 THz). In this figure, we set the transmission window with 0.62-0.79 THz, 170 GHz wide. Three common power allocation pulses are proposed here, rectangular-RCP, Hann-RCP, and Gaussian RCP. Rectangular-RCP is a flat pulse and cannot resist the path loss and noise efficiently. The Hann-RCP mentioned above performs better in short range

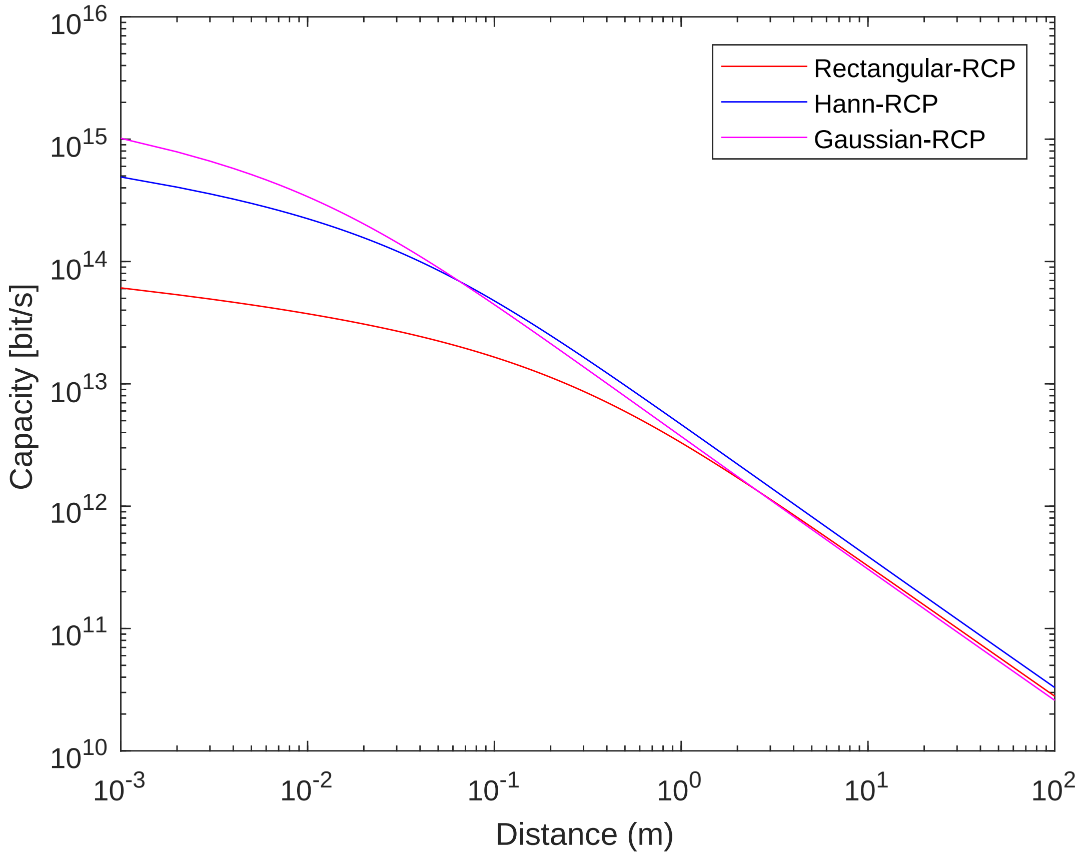


Fig. 3.2 Capacity as a function of the distance for three different power spectral densities

than Gaussian-RCP. The Gaussian-RCP is not as sensitive as the others when the distance increases. Other pulses such as Sinc-RCP and carrierless IFFT pulse also have specific applicable systems of terahertz vehicle communications.

3.2 Terahertz Autonomous Relay

Similar to the study result in recent paper [3], we observe the propagation of THz level signal at 1 and 10 meter decreases by 80 dBm and 110 dBm, respectively. THz communcation systems are proposed to achieve a 100 m link with directional antennas and shorter than 10 m link with omnidirectional antenna [34]. In order to extend the available transmission distance and enhance the resistance of environmental change, our proposed autonomous terahertz level relay (ATLR) becomes a suitable solution for vehicles. Since there is no universal standard for such short range communications, Tx can use the whole band to enhance the spatial reuse, which keeps a large volume of throughput in each interval. Besides, through relaying, it changes the lower penetration of NLOS communications into more stable LOS links. Moreover, it provides moderate length links which have the same distance with Wi-Fi but the bit rate is higher by 5-6 orders.

3.2.1 ATLR Algorithm

However, terahertz transmitters and receivers have to suffer the constantly changing of propagation and noise when traveling on the road, which makes the relay point hard to select. Therefore, we propose an autonomous terahertz level relay (ATLR) algorithm to find the best relay position. The main idea of the optimization strategy is to detect the channel quality of each position in 2D representation and find the optimized point. In this scenario, we divide the whole space into very small blocks and detect the channel quality in each of them. However, under complicated traffic conditions and a l^2 plane area for detection (l is the defined side length of the plane for an independent relaying scenario), it is impossible for the relays to collect all the information from every space block. Our algorithm calculates the optimized relay point with relatively less movement when considering the advice of possible automatic driving routes.

Fig. 3.3 is an example of our proposed ATLR algorithm. Car A and B are autonomous vehicles with the mentioned support of 3D image detection system. Their autonomous systems advise them to advance following the blue line. A candidate relay Car R is by the side and has the similar self-driving route (noted as the red line). The car B sends a request to relay R, then R uses the mentioned carrierless IR pulses to detect nearby cars. Notice that

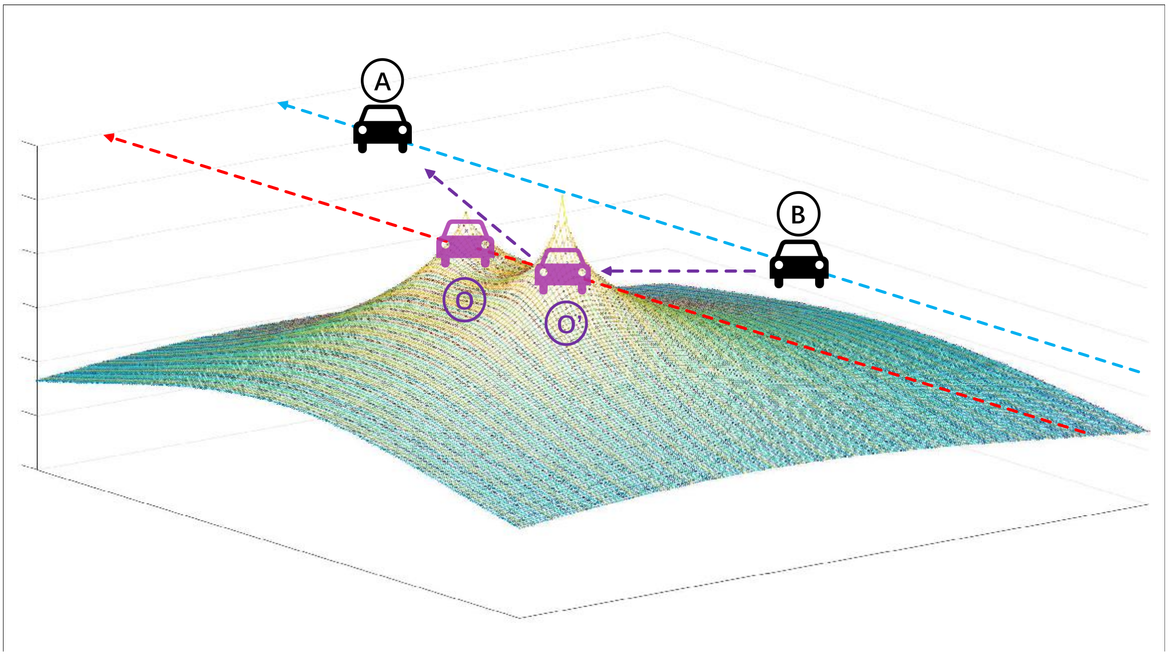


Fig. 3.3 An example of autonomous terahertz level relay algorithm (ATLR). Car A and B is the sender and receiver. Pink car tends to find an optimized position using ATLR for relaying A and B

the conventional RSS method or the state of the art LiDAR (Light Detection and Ranging) sensors can also be embedded on an autonomous car for positioning. With the awareness of surroundings' relative distances, relay R moves to the temporal optimal position O_1 as Round 1, which calculated by ATLR (the position noted as O in the figure). When moving to the suggested position, R detects the quality of the channel in each space block and stores in a channel quality matrix Q. After holding at the temporal optimal position, ATLR recalculates the optimal position as O_2 (noted as O' in the figure) based on the relay quality value $Q_{n,m}$ we defined. This value is stored in the channel quality matrix Q, where we have

$$Q_{n,m} = (Tx_{n,m} - Rx_{a_1,a_2}) * (Tx_{n,m} - Rx_{b_1,b_2}) * L_{n,m} \tag{3.3}$$

where $Tx_{n,m}$ is the signal quality of the transmitting antenna and $Rx_{i,j}$ presents the receiving antenna. We use $L_{n,m}$ to indicate recommendation rate of the route calculated by autonomous driving system. It depends on the traffic condition dynamically updated by the road understanding system, which the risk areas are eliminated. This setting can also avoid other nearby vehicular communication systems reasonably. When moving to O_2, ATLR recovers $Q_{n,m}$ and calculates O_3 at the end of Round 2. After k rounds, the relay vehicle fine-tunes itself to the stable position O_k. Since the relay can only monitor channel quality blocks passed by, the moving distance of each round is monotonically decreasing, which finally stops in a specific block. Through the process above, car A and B keep the stable and ideal terahertz channel through relay R.

3.2.2 Autonomous Multiple Relaying

Under the condition of certain dense traffic, there may be more than one candidate relay available in this scenario. The multiple relays are introduced here to further extend the limited range between car A and B. Firstly, through position sensors, the ATLR system estimates the needed number of relays to establish multiple hopping. Similar to our previous study [27], we propose a multi-deviation transmission protocol to improve the channel quality aggressively. The basic idea is that by setting common id as a priority (such as MAC address, user id, etc.), the ATLR produces a map with suggested position $O_{0,i}$ for the first priority Relay R_i. The relay R_i firstly moves to its optimal position $O_{0,i}$ then a single relay $\{R_j|id(R_j) < id(R_i)\}$ starts to collect the channel information when moving to its $O_{1,j}$. Then relay R_i moves to its optimal position $O_{1,i}$ and finishes the first round. Notice that this process can be applied to more than one relay, which can avoid priority or collision problems. A thoughtful setting of the priority in a specific scenario can even improve the efficiency of the whole system.

3.3 Enhanced Autonomous Driving

Autonomous driving has become a hot topic in recent years. Many amazing achievements and applications have emerged. However, these researchers mostly focus on the correctness defined by human beings. Some pioneers consider the autonomous driving system should perform better than human driving in an all-round way, which requires more remarkable perceptibility and reaction in emergency situation. However, even if the autonomous system can nearly understand the visual information as well as human beings, it can never surpass human being's perception. One of the advantages of vehicle network is the information sharing ability. Conventional vehicles share information and traffic condition through DTN or RSUs [35], which only provide size limited message transmission. The application of vehicle terahertz band communication will bring new opportunities to the autonomous system, to see what human drivers can never see and to learn what human can never learn.

3.3.1 Performance Evaluation

Next, we conduct the simulations in a scenario of autonomous vehicle networks. The scenario contains a 6 six lane expressway, where 3 lanes in each side. The width of each road is 3.5 m and the total length for observation is 200 m, where 12 RSUs (Road-Side

Unit) are established at the road side. These RSUs contain the equipment of various vehicle communication systems and imaging sensors, providing 3D imaging and HD videos for autonomous vehicles nearby. The antennas of mmWave system and THz system are both considered as omni-directional antennas. If one transmitter sends information to the receiver of another autonomous vehicle, the receiver cannot collect others' information using the same frequency channel. If the channel collision happens, vehicles with higher priorities can establish the link. The problem of priority can be solved by many ways, such as importance of data, vehicle types, etc.

In Fig. 3.4, it shows the simulation results of average number of links using different wireless communication systems. One link represents two or more connected autonomous vehicles. All the vehicles obey the rules of the road and minimize the dangers of driving. Three basic transmission modes are tested here, gigabit communication, mmWave communication, DSRC intelligent transportation systems (ITS). The Gigabit communication (noted as THz in the figure) can approach 2.5 Gbit/s in real deployment within 10 meters while mmWave can keep 0.1-1 Git/s using different approaches. IEEE 802.11p based DSRC has the rate of 3-27 Mb/s with the communication range of 30 meters. RSUs are used in both mmWave and THz to provide additional road information in case of no vehicle's part, noted as $RSU\&mmWave$ and $RSU+$ respectively. Since conventional communication technologies such as DSRC can remain stable links only for images, we consider the captured HD video data can only be used by mmWaves and THz technologies. $ATLR+$ represents the links established by three vehicles, where one is the relay that builds bridge using different channels.

With the increase of autonomous vehicles, the average number of links is growing. However, the slops of all the communication systems decrease due to the interference of other links. The DSRC keeps keeps stable standard due to the relatively long communication range, where the collision problem is obvious. The number in THz system is larger than than the number in mmWave due to the purer communication environment in short range. It is more flexible to establish reliable links in high density. Never than less, by the supporting of RSUs, there are more information shared in the air to improve the driving decision system. The $ATLR+$ is used to establish multihop links, and more links are established due to the higher density. It collects more information from different vehicles to improve the accuracy of road condition recognition system.

Currently, higher frequency transmission systems such as Thz bandwidth may be not fully developed to deal with complicated traffic conditions. However, the advantages of higher transmission rate and attenuation help create more efficient transmission environment in short distance. Therefore, a hybrid transmission system using various communication formats to handle various situations is one of the future tend. It minimizes the interference

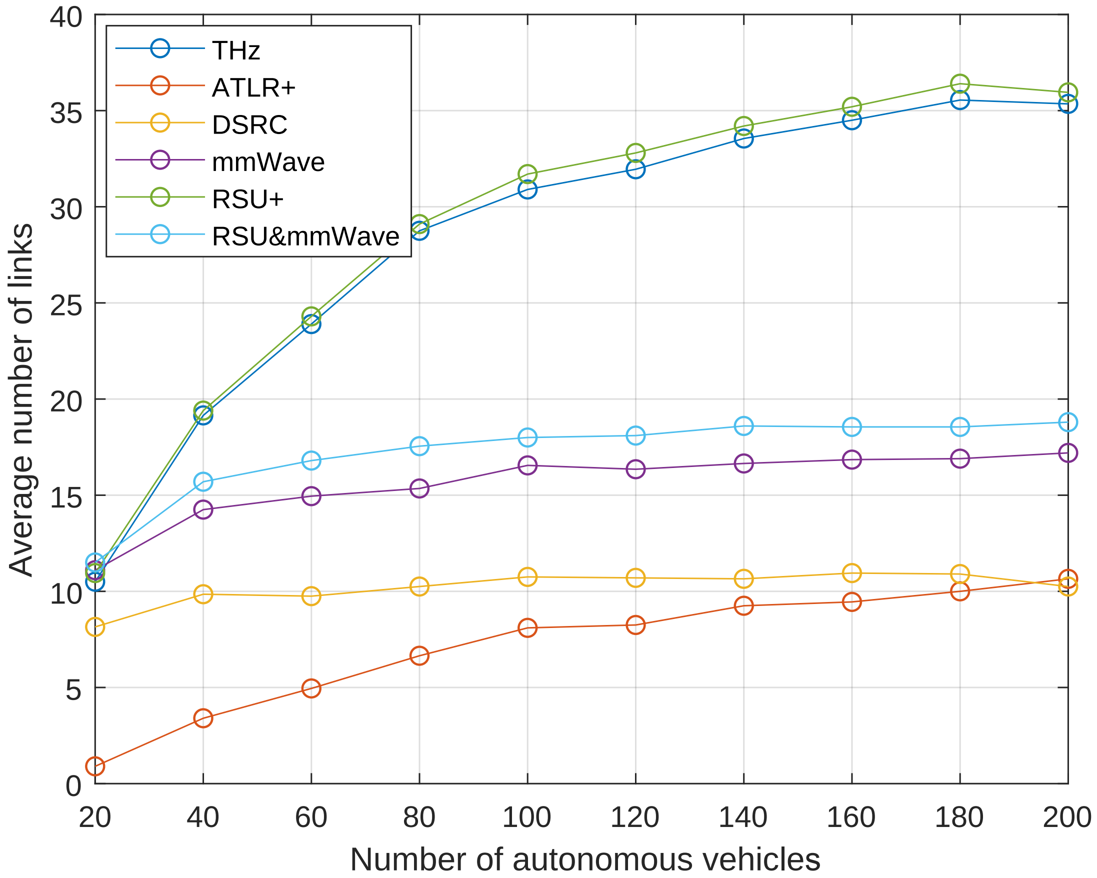


Fig. 3.4 Precision comparison in autonomous driving through different technologies

from different channels. A hybrid approach may adapt more single or multihop links to heterogenous networks, which provides more flexible options for autonomous vehicle communication.

3.4 Conclusion

In this paper, we study the impact of terahertz band towards vehicle networks. THz band communication relieves the spectrum scarcity and capacity restriction of the existing communication systems. In short range, the THz link can be considered as a transmission window with almost 1 THz, which supports the growing real-time data transmission. Moreover, we develop an autonomous terahertz relay algorithm called ATLR which gives the advice of ideal relaying position to bypass obstacles and avoids stronger NLOS fading. Finally, we apply this algorithm in autonomous vehicle network. It helps create more flexible communication environment and provide more traffic information to nearby autonomous vehicles.

Actually, not limited to autonomous vehicles on the road, terahertz band can be widely applied to all mobile nano-cell networks, such as delivery drones or UAVs (Unmanned Aerial Vehicle). Different kind of 'vehicles' has very different ability to adapt communication environments. These 'vehicles' support the exploration of missing details, which helps human beings and human made machines understand the known world more smartly.

Chapter 4

Social Taxi Sharing: A Cyber-Physical Approach for Efficient Urban Transportation Service

This chapter describes a real time and real environment solution to meet with the experience issue in urban car sharing services, where the user-oriented taxi sharing service using social network is considered in this study.

In this chapter, we propose a cloud-based taxi-sharing system, which provides real-time taxi sharing services both considering the practical travel needs and QoS in the journey. The purpose of this service is to minimize the travel cost through sharing calculated schemes reasonably and quickly and improve the sharing experience as much as possible. We model the citywide level map into blocks and indexed, in order to search for the potential paths of ride-sharing dynamically and intelligently. Meaning while, we explore the paths among communities to find suitable companions as well. Finally, as well as the establishment of the mobile application and online service, we also conduct a large-scale simulation to evaluate the performance with others. The result shows that the proposed solution could make the whole taxi-sharing system less calculation complexity, higher sharing QoS, and most of all, keeping the minimum cost of travel cost within 1.16% increment than distance optimal algorithms.

The main contributions are summarized as: social network relationship is introduced into the taxi resource management when limited taxi resource meets with the rush hour. The community structure of a social network is taken into consideration when scheduling socially connected passengers in one taxi; An optimization problem is formulated and calculated using our proposed 3-step process, including community detection, real map taxi searching and social network based heuristic algorithm called SONETS. It searches for an optimized solution to make a cost efficiency, fairness, and humanized service designing; New smart device application called 'Social Network Taxi Sharing' is designed and applied in the real environment. Furthermore, the large-scale simulation is deployed based in citywide, using real user information and cartographic information.

The remainder of this paper is organized as follows: in chapter 4.1, we detail the proposed architecture of online taxi sharing service. In chapter 4.2 and 4.3, we work on the taxi allocation problem and develop the optimization algorithm called SONETS. Performance evaluation is presented in chapter 4.4, and the conclusion is in chapter 4.5.

4.1 Sharing Service Design

This section presents the motivation and structure of the proposed taxi sharing oriented CPS. Cyber-Physical System (noted as CPS) is essentially a multi-dimensional intelligent technology system. Our work is based on big data, on-line network services, and massive computing. Through core intelligent vehicular sensing, traffic analysis, schedule optimization, and route coordination, the proposed CPS will be used for service computing, communications, and control (3C), in order to achieve in-depth integration of the network space (social network, online service) and physical space (real map, taxi service), which changes the lifestyle and improve the life quantity. Due to the rapid development of 'cyber', every person may carry one or more smart devices for online services, and forms various networks, e.g., social networks, communication networks, location-based online social networks, etc. Then, we consider the situation that is sharing potential passengers all have strong or weak social network connections with each other. They have smart devices which access the Internet and are willing to reach their destinations by sharing a taxi [47]. The sharing system provides the taxis and sharing arrangement service by working on the coordination of different travel plans, and social network infects between passengers.

Next, we discuss the feasibility of this scenario. There is a period in every day that potential consumers suddenly need many taxis which we call it taxi-needed peak time: For example, at lunch or closing time, a mass of potential users in the same social community of company are waiting at the front door of an office building for taxis. Usually, sharing taxis by street hailing is a commonly solution in such a taxi resource-limited situation [25], but still need to wait for taxi's arriving; Actually, these potential passengers may share the same community if they share similar activities or work at the same place, and the inner connections are strong enough as well. With the trusted relationships, they are willing to share the same taxi if they have similar paths [57]; Other potential passengers such as students are willing to share because they are a group of people who are sensitive to the cost and desire more social interactions [18], and most of, they live in the same camp.

Our new social network based taxi service is proposed for the above scenario. Through the coordination of time, taxi and cost, it is designed to schedule potential passengers whom others may be interested, through which we consider the social network relationships. The social network is an online service for establishing or reflecting the social relation of people who have similar interest and activities [54], and location-based networks could even show the real daily travel needs in a city. Similar social features are the reasons why they connect to each other, as well as establishing a big social community. It helps us find the persons who may share the same lifestyle and give out recommendations to other consumers [41]. Therefore, a social network connection is a beneficial recommend status for choosing the right sharing group. For example, graduating from the same school is a reference factor that makes the system tend to arrange them in the same groups, which can help satisfy or even cheer up them easily during the same trip [30].

4.1.1 System Demand

For replying to the above concerns, we design a framework of the proposed system with demands below:

1. Users can update their taxi sharing travel plan with a mobile device through an interactive user interface.
2. All taxis are monitored by the system so that all potential passengers are in the same queue and experience same standardized service.
3. A solution about the detail information of the routes and passengers is provided for each taxi, which takes current passengers' travel plan and social connections into consideration.

4.1.2 System Structure

Fig.4.1, it represents the architecture of our proposed social network optimized taxi service. This architecture contains three parts: user part, operation part, and data part. User part contains potential passengers who desire for taxi sharing and all taxi resources. Operation part schedules the detail operations in the user layer. The data part provides all necessary data asked by operation part.

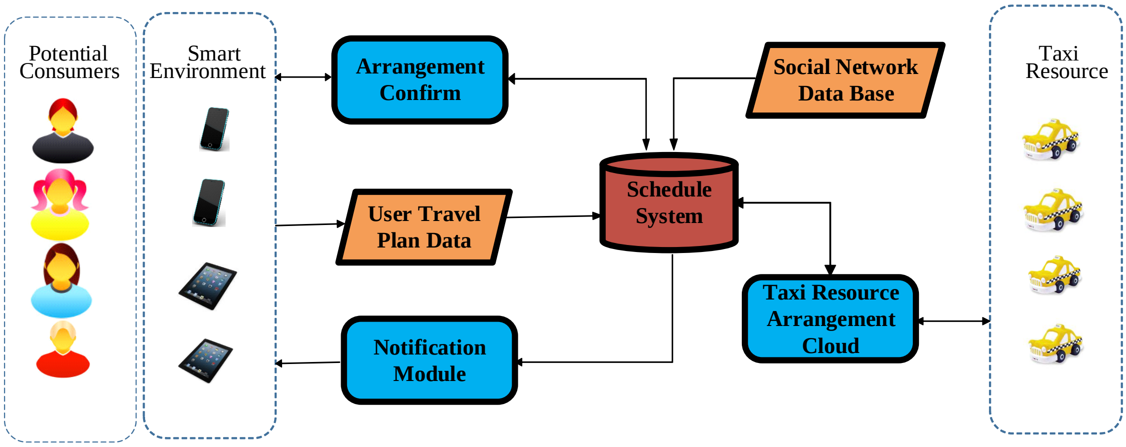


Fig. 4.1 Structure of the social network based taxi sharing service

Taxis Resource: One taxi driver uploads their idle time and location. Then, the system considers it as taxi resource at one specific starting point.

User Interface: The module includes all the necessary sharing plan information sent by the potential passengers. When users planned their trip and needed a taxi, they submit their plans to via the user interface. Before real departure time, all the untimely information submitted by potential passengers are stored in this module, and be updated into the global request queue dynamically. When the time comes, it will be sent to the Schedule System Module. Generally, it contains four functions: account management, personal data management, result informing and social network data authorization.

Taxi Resource Arrangement: This module is to monitor all taxi resource and inform the sharing arrangement system.

Schedule System Module: This module works out the schedule of the entire arrangement of every taxi. This module calculates the solution about how to allocated potential consumers in different taxis. We use a social network based algorithm (SONETS) to complete the passenger allocation. The input of the processing is based on the data of the travel plan in User Travel Plan Data and Social Network Data Base. Finally, it sends the allocation schedule to the Arrangement Confirm Module for notification.

Arrangement Confirm Module: It pushes arrangement results from the Schedule System module to users. The pushed information includes the distance of the whole trip, adjusted departure time(one taxi one departure time), number of passengers in a taxi and cost. After all the informed users in one taxi confirmed the arrangement, the driver considers this schedule is established. If not, we consider the consumers give up their chances, and delete from the queue. The other request is sent back to Schedule System module for a new solution.

Social Network Relationship Database: This module stores user social relationship which based on the social network (e.g., Facebook, Twist and YouTube). Data mining and network analysis are processed by the cloud dynamically, and the users are classified into community settings. This community definition will be used as the standard whether the two potential consumers' interrelation is tight enough. We consider the tougher the social interrelation is, the more significant chance they should sit together.

Notification Module: It pushes confirmed results from sharing arrangement module to the specific consumer. The difference with Arrangement Confirm module is that it combines the specific travel information from both sides, which contains the travel information about the current positions of taxis and passengers. It enhances the recognition for people. Notice that a reminder function is also embedded in the application to guarantee consumers' attendance.

Taxi Resource Arrangement module: There are two primary functions for this module, embedded with a driver-oriented interface. One is to monitor all the taxis in citywide and inform the Sharing Arrangement Module the available taxi resource. Another is to push detail information of the accomplished travel plan to drivers, including destination, the drop-off place of each passenger, identity information, departure time, etc. Notice that this module connects to the taxi company, where the taxi service data can be further reused for meaningful data analysis.

4.2 Map Processing for Cities and Users

In this section, before giving out a practical system model, we firstly pre-process the city map that divided by blocks and makes all the travel requests in indexes for different blocks. Then, we propose a system model in order to describe the travel plans numerically and calculate the cost on the city map. Finally, we discuss how to search for suitable travel companions on the social network map.

4.2.1 Map Planning

For each passenger, their start and destination points updated to the system may not be the final positions picked or dropped, since most of the taxi drivers could only choose a near drop off point on the road [25]. Therefore, we assume that all the passengers in one block could be dropped off at the edge of a real-world block. For a taxi, if it can cross by the edge of one block, we consider it can drop off all the passengers whose destinations are included in this block. The similar idea is proposed by [10] early, which used in passenger recommendation in local areas.

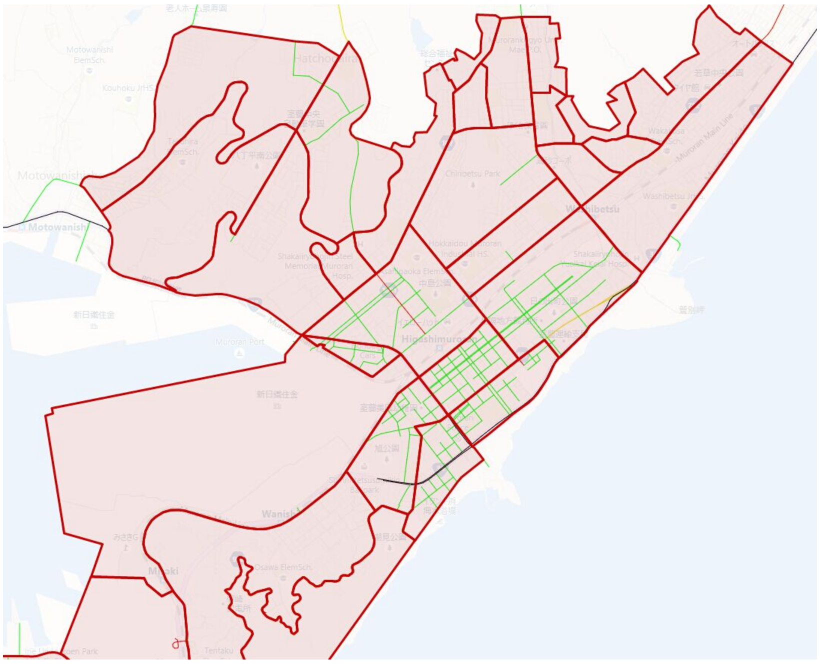


Fig. 4.2 Muroran map divided into blocks

Step-1 Map Division: Fig.4.2 is a map of Muroran which has been divided into many blocks using map street data. The main purpose of this step is to divide passengers' travel plan geographically. We select three types of roads as our edges, highway, trunk road, and minor road. Every block is a closed polygon that consists of several roads and vertexes. Since Muroran City is a port city that faces the Pacific and has mountains behind, there are many mountain roads which couldn't cross each other and form blocks. Therefore, it needs extra manual work to form new blocks, which presents new destination areas. For each passenger, his destination is only contained by one block in this map. One block contains a collection of passengers who share the same edges. If one taxi crosses by the edge of this block, the passenger whose start or destination point belongs to this block is considered as a good candidate for sharing, and there is no detour for dropping off or picking up. Notice that if two parallel roads are too close to each other (within 5m), the first one forms a block, and another one will be ignored during the block division processing.

Step-2 Block Extension: This step is to meet passengers' habit of getting on and getting off. On the outside of a block, there may be several users nearby and next to block, where dropping off them at the same edge is the best choice. Besides, the road has a fixed width with two sides for parking. Therefore, it is practical to extend boundaries for each block to expand the scope of potential passengers. Fig.4.3 is an example of how these blocks extend and cover each other. Each new block contains the collection of original block members and other users who are close enough. The edge is extended by 1.5m for minor road, 2m

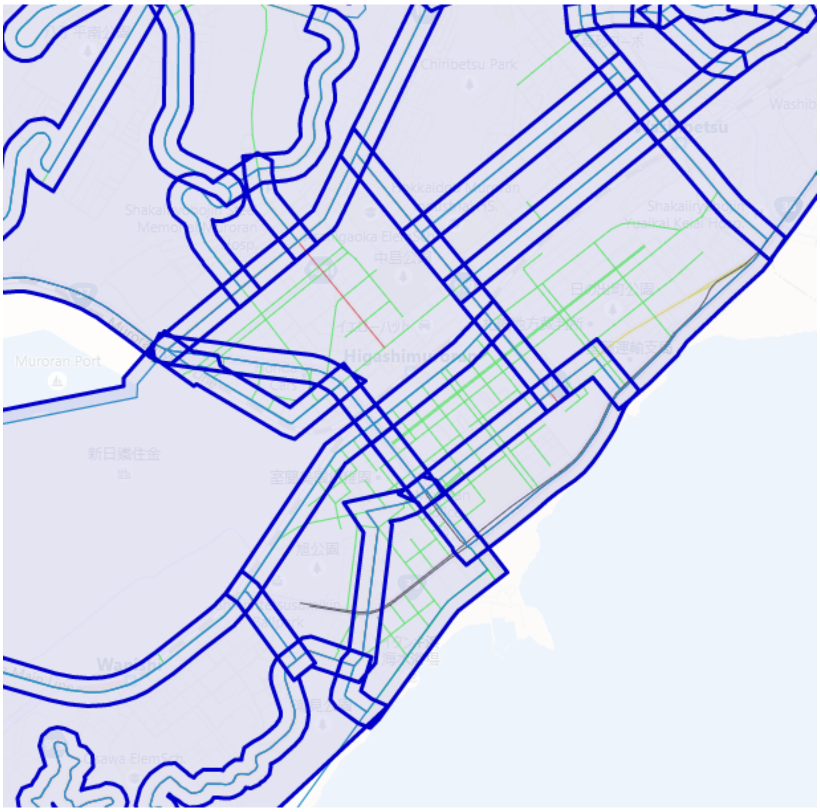


Fig. 4.3 Muroran map contains extended blocks

for others, based on the road infrastructure information. We select the center of a polygon and push the border parallel to the outside for 1.5m/2m to form a closed new polygon. By processing the map, blocks are divided by roads, and contain common users in both that are 'close enough'. The 'close enough' means the node is outside of the specific block before the extension and included by this block after the extension process. Notice that, the different colors in the figures are to distinguish the status before and after the extension (in blue and red respectively), and make a clear vision since it will be hard to recognize when map zoom out.

Step-3 Taxi Patrol Cover: The purpose of this step is to provide a rough passenger dataset in taxi's patrol path. From the start point to the destination, one taxi shall cross by many blocks. The users belong to these blocks are appropriate for sharing candidates geographically, because it takes fewer detours and time for dropping off. Therefore, we define a "cover" here to indicate the taxi's patrolling area, which helps select the set of suitable passengers geographically. As in Fig.4.4, The arrow indicates the direction of the taxi. Generally, one taxi crosses by several blocks. Each block contains a set of potential passengers (noted as the circles), and each set is a subset of all potential passengers collected

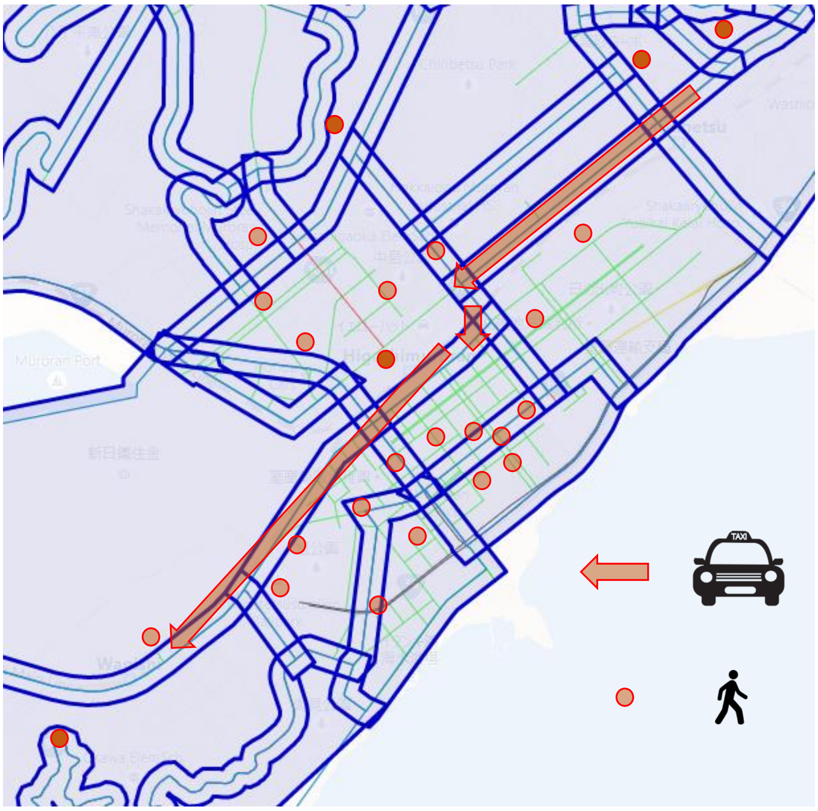


Fig. 4.4 One cover contains several blocks which refers to sets of potential consumers

by the corresponding path. In other words, when we establish a path, it generates a union set combined with several subsets, which contains passengers indexed by each block.

Significance: Through the method of map planning, we narrow down the searching scope for available passengers geographically and improve the calculation speed. It reduces to a sub-problem from the whole user data set, where we develop a heuristic algorithm to find the solution in the next section.

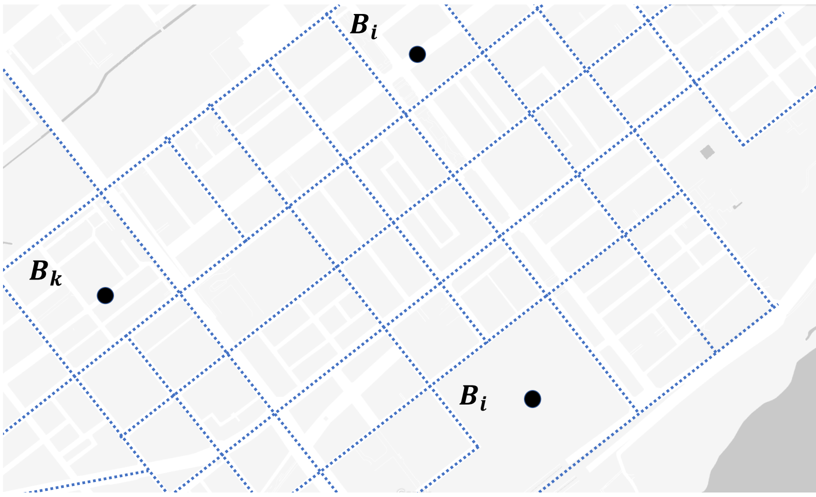
4.2.2 Index of Blocks

After the block division process, each block can hold an index which describes the basic feature of this block and active nodes in it (taxis, passengers, destinations, etc.). The purpose of the index is to reduce the calculation complexity and speed up the route searching process. As shown in Fig.4.5a, in each block, one node is chosen as the anchor node for this block, which is the geometric center for the polygon. Notice that other center node selection method like the central intersection also can be taken into consideration. Due to the length limitation and incomplete map information, we use an anchor node instead to make the description easy to understand. Then we can calculate the distance, travel time, and fastest path between blocks on the map through various methods.

To make a practical system, we use the travel time as an important indicator for travel planning, since it can be dynamically updated by the online service and has the statistical significance to reflect the traffic condition in a specific period (as we know, traffic flows on the morning rush hour is totally higher than wee hours). It can be estimated through average travel time by different taxis, which updated by each taxi after each contract is finished. Besides, the travel time between blocks can be used to filter out those blocks that taxis cannot reach in time within $O(w)$ computation time, where w is the total number of blocks. For example, in Fig.4.5b, the matrix C represents the statistical result of travel information in each block, and each symbol B_{ij} indicates a directed travel information (e.g. travel time) from i to j. It is possible that B_{ij} and B_{ji} have a big difference, which depends on the direction of the overall traffic flow in the citywide (or different period). The advantage of this matrix is that the travel time and distance are independent of the real-time path-finding algorithm, which increases the fault tolerance and lateral compatibility for other up to date navigation systems.

As shown in Fig.4.6, the travel information of each block B_i contains three parts: a travel distance based block list $B_j.d_{ji}$; a travel time based block list $B_j.t_{ji}$; a taxi arriving list $Taxi_m : t_i$. The travel distance $B_j.d_{ji}$ indicates the actual travel distance from arbitrary block B_j to block B_i in its index and they are ordered by ascending. The travel time $B_j.t_{ji}$ is also ordered by ascending based on the travel time from other blocks. The taxi arriving list $Taxi_m : t_i$ is a list that indicates the planned arriving time by scheduling system or other stand-by taxis, and this list is in ascending sort as well (taxi stops at the original block B_i is considered as 0). The spatial distance (travel distance) and temporal distance (travel time) can be directly inputted from the block relation matrix C. The designing goal of these lists is used to search for the potential combination of travel plans. For instance, if one passenger wants to search for taxis in nearby blocks within 10 minutes, the system can filter those taxis in blocks that $B_j.t_{ji} > 600$ s. The system will update these lists periodically after new information is uploaded by each taxi. Notice that the extension of blocks will not create the overlapped problem in remote blocks (start from different blocks) because the real arriving time is estimated from the actual start point and uploaded based on the real road path.

The taxis list $Taxi_m : t_i$ contains the taxi ID that scheduled to arrive in the future, and the ID is acceptable to be listed in both nearby blocks it crosses over. The actual arriving time is tagged with a stamp t_i, and sorted in ascending order. These timestamps are updated immediately after a single schedule is established, and they will be moved at once when the taxi passed by. Meaning while, if a new schedule is confirmed (whether it is the new



(a) Indexed block map

Block ID

0	B_1	B_2	B_3	B_4	......	B_n
B_1	0	B_{12}	B_{13}	B_{14}	......	B_{1n}
B_2	B_{21}	0	B_{23}	B_{24}	......	B_{2n}
B_3	B_{31}	B_{32}	0	B_{34}	......	B_{3n}
B_4	B_{41}	B_{42}	B_{43}	0	......	B_{4n}
......	......	......	......	......	0	......
B_n	B_{n1}	B_{n2}	B_{n3}	B_{n4}	......	0

(b) Directed travel information matrix

Fig. 4.5 Formulate the blocks by index and calculate the travel information between them

trip combination or individual plan), the newly created timestamps will be inserted to each block in the right order. In the real taxi sharing service, it is impossible that the taxis update continuously the GPS coordinators all the time to show the precise in-and-out time in each block. However, it can keep updating with a series of discrete GPS nodes, and the nodes appear in the right block are indexed in these blocks.

4.2.3 System Model

We consider there is a status set N of users $\{1, 2, \ldots, i_{max}\}$ of potential users who are willing to share a taxi. A set status M with taxis $\{1, 2, \ldots, j_{max}\}$ are waiting at the blocks. Then,

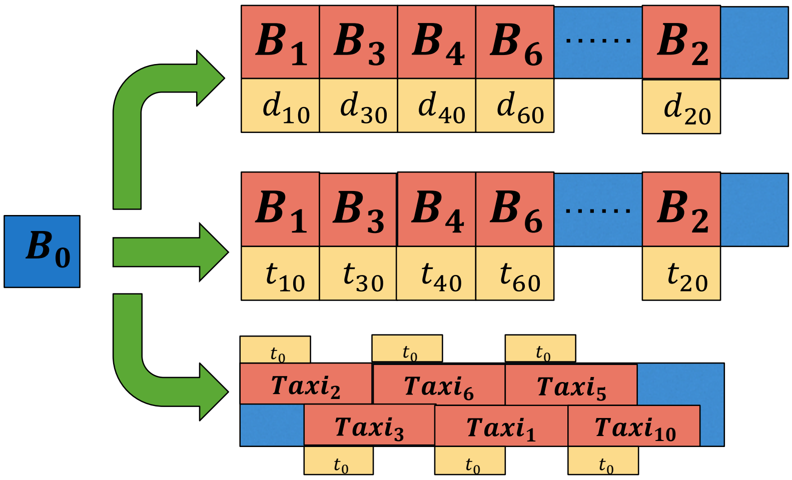


Fig. 4.6 The index of a block that contains the list of taxis and travel information from other blocks.

we consider the taxi sharing peak time is a specific period divided by time slots, noted as $T = \{1,2,\ldots,t_{max}\}$.

Time Frame: Since taxi sharing peak time often appears in a specific period in daily life, similar to [48], we consider this fixed period with same length slots. For a fixed length period, the system creates dynamic queues both for candidate consumers and taxis. Scheduled passengers and taxis will be offloaded from the queues.

Destination: For arbitrary user $i \in N$, we define an array $X_i = (longitude, latitude)$ to indicate the travel plan of user i. Z_i is the distance between the destination and start point. We define

$$Z_i = D(|X_i - O_i|) \tag{4.1}$$

which denotes the distance function of the direct path by a taxi from starting point to the destination. The result can be solved by inputting vertexes and edges from the open street map data and applying with common shortest path algorithms. Notice that O_i indicates the coordinate of starting point and $Z_i \in [Z_0, Z_{\max}]$.

Cost: For arbitrary taxi $j \in M$, the total payment C'_j in a taxi is formulated as,

$$C'_j = P_0 + (Z'_j - Z_0) \cdot P_1 \tag{4.2}$$

where P_0 is the starting fare and P_1 is the cost per mile. Z'_j means the total distance taxi j moves. Z_0 is the distance of starting a course. Without loss of generality, the proposed system calculates the cost for each passenger based on the proportion of distance they experienced. It can reduce the conflicts on cost segmentation[14]. Since passengers still need to endure

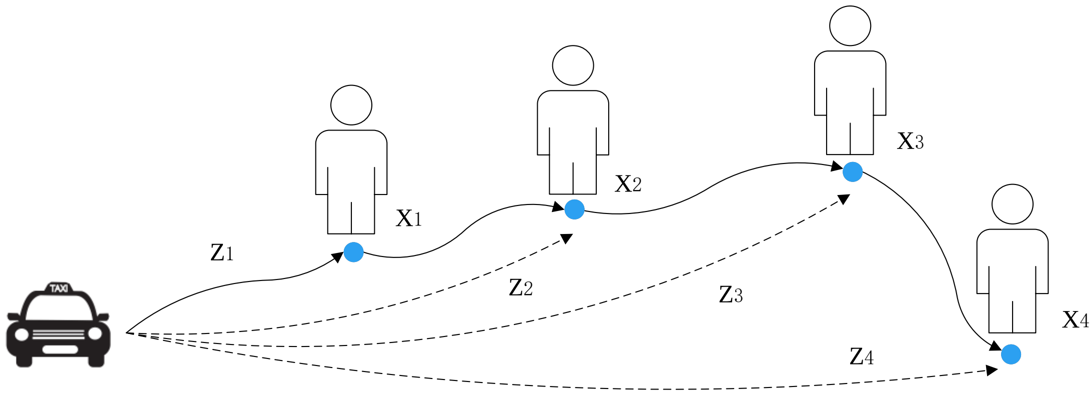


Fig. 4.7 Routes begin from original point to each passenger's destination

the waiting time and detour of dropping off [64], the proposed recommendation system need to handle this problem smoothly and flexibly, which we discuss in the next section.

Since the original route is established by the first passenger, who is also at the head of the queue, it is fair that all the passengers undertake part of the fee from the start point. Therefore, we define our distance-based cost division method is calculated from the start point of the first passenger to the destination of each. Through the sharing, the passengers behind the queue have the chance to skip the queue and take the taxi. Moreover, it forces passengers to use the sharing system efficiently. For example, one passenger's start point is near first passenger's destination, and a taxi should remain seats empty for a long time.

Travel cost is referred to the proportion of each passenger's travel distance. The basic description is based on Fig. 4.7. Here, for each passenger

$$Z_{j,b} = \sum_{a=1}^{b} D(|X_{j,a} - X_{j,a-1}|), \text{ where } X_0 = O_0, b \leq a_j \tag{4.3}$$

where a_j is the number of passengers in taxi j. Obviously, we can infer the actual distance of taxi j's path Z'_j. Their respective cost is divided from the total taxi fare C'_j, which using the proportion of their owe riding distance. For the no.b passenger, the cost $C_{j,b}$ can be formulated as

$$C_{j,b} = \frac{Z_{j,b}}{\sum_{y=1}^{a_j} Z_{j,y}} \cdot C'_j \tag{4.4}$$

The basic reason that one passenger wants to share a taxi is the cheaper cost. Our payment division method should not break the base line of sharing. Therefore, the passenger b's payment $C_{j,b}$ can be stated as,

$$C_{j,b} \leq P_0 + (D(|X_{j,b} - O_{j,b}|) - Z_0) \cdot P_1 \tag{4.5}$$

for the b-th passenger in taxi j.

Economic benefit: The system needs a quantifiable concept to measure the economic benefit. We define a variable EB_j to indicate the economic benefit by sharing. Comparing with taking a taxi alone, we have the economic benefit (EB) for taxi j here is

$$EB_j = \sum_{b=1}^{a_j} (D(|X_{j,b} - O_{j,b}|) - Z_0) \cdot P_1 - (D(|X_{j,a_j} - O_{j,1}|) - Z_0) \cdot P_1 + (a_j - 1) \cdot P_0 \tag{4.6}$$

4.2.4 Sharing Social Communities

After finding the available taxis in nearby blocks, next we discuss how to search for potential passengers to improve the quality of experience during the journey. Since communities in the social network could represent the real social interactions, social groups, and location-based activities, we collect location-based social network information from these users to establish an integrated share schedule through a social network map. The distance on the social network map is an important factor to indicate the relationship strength between different individuals and communities. This idea has been applied in many fields: relationship of paper authors can be effected on the citation network; the possibility of appearing in the same place shows how often the two people meet with each other.

As shown in Fig.4.8, a general social network can be formulated as a network of communities that there are more inner connections in one community and less global connections between different communities. For a general social network dataset, there is a weight to describe how close the pair of vertices is. One vertex can be connected to multiple vertices through edges, and each edge holds different weight. The traditional method to draw a social network is that we can start with the edge which has the strongest weight. Then, we add edges pair by pair in the descending order until the weakest edge. Through the edge adding process, the network graph shows nested structures in specific areas, which could be considered as communities. For each nested structure, it also can be formulated as a tree structure to emphasize the strength of a relationship. For example, as shown in Fig.4.9, the first level represents the first edge that is leading to the vertex classified into this community (in most of the cases, it is the first edge added on the network for this vertex). A slice through any level of edge would lead to smaller communities or groups. Edges in a higher level

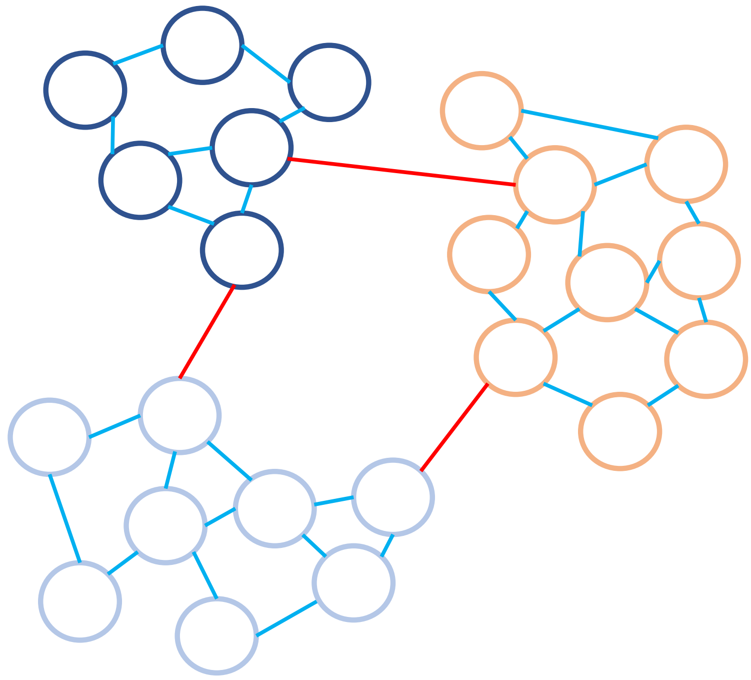


Fig. 4.8 Structure of the social network community

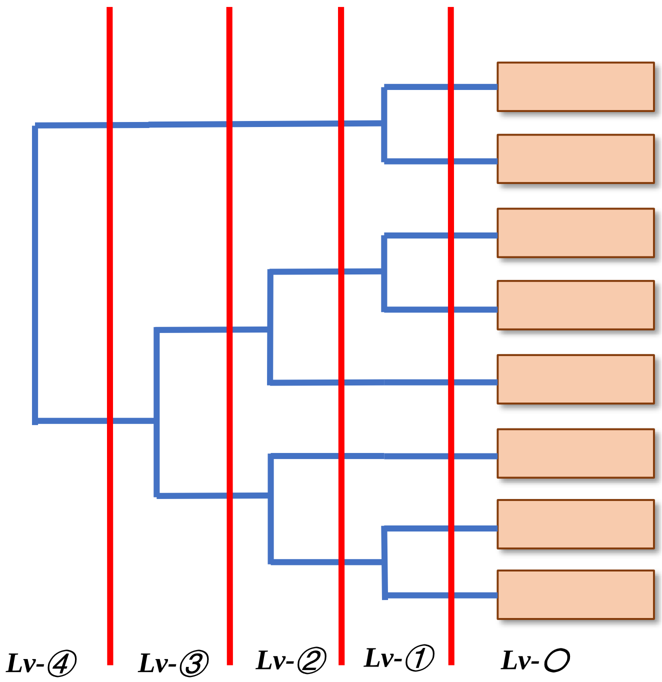


Fig. 4.9 A tree structure of social network to show the robustness of each link

show more important in establishing these communities. This structure is also similar to dendrogram in hierarchical clustering.

Community Detecting Model

Based on the above observation, we can discover that some edges (high-level edges) have played an important role in connection and differentiated communities. In another word, a pair of vertices belongs to different communities have to pass at least one of these edges to establish the shortest path. In traditional methods, the system estimates the importance of each edge during the edge adding process, which could help establish a broader community. On the other hand, these edges are also the key edges to divide vertices into groups. In the view of a global social network, those edges are the least central edges for each community. Therefore, instead of adding edges with the most substantial weight to construct communities, we classify the community members by removing the most 'busy' edges between communities. The essential idea is that by removing the most important edge between communities, we can discover a community that they have only inner paths after the former most "busy" edge is removed.

In order to find which edge is the most important between communities, we introduce a concept of betweenness in previous works. Usually, the vertex betweenness is used to measure the centrality and influence in social networks. A node with high betweenness plays a more critical role in the information flow over the social network. This specific betweenness is estimated by calculating how many shortest path have passed over this vertex. In this paper, instead, we calculate the betweenness for each edge. If a vertex belongs to one community and wants to connect with outside vertex, it has to choose one outer group edge to reach the vertex in other communities. These edges connecting these communities always hold higher edge betweenness and should be firstly removed in descending order. Fig.4.10 shows the process of community detection. The community detecting process is summarized as follows:

1. Find the shortest path for each pair of nodes.
2. For each path, add one count on the edges it passed by. If there exist two shortest paths with the same length, all the edges covered by these paths will be counted once.
3. Find the edge which holds the highest betweenness and remove it.
4. Calculate the betweenness effected by the above removing.
5. Repeat the process until vertices in one community cannot contact with outside.
6. Store one less member group as a community and repeat from step 1 until all the communities are found.

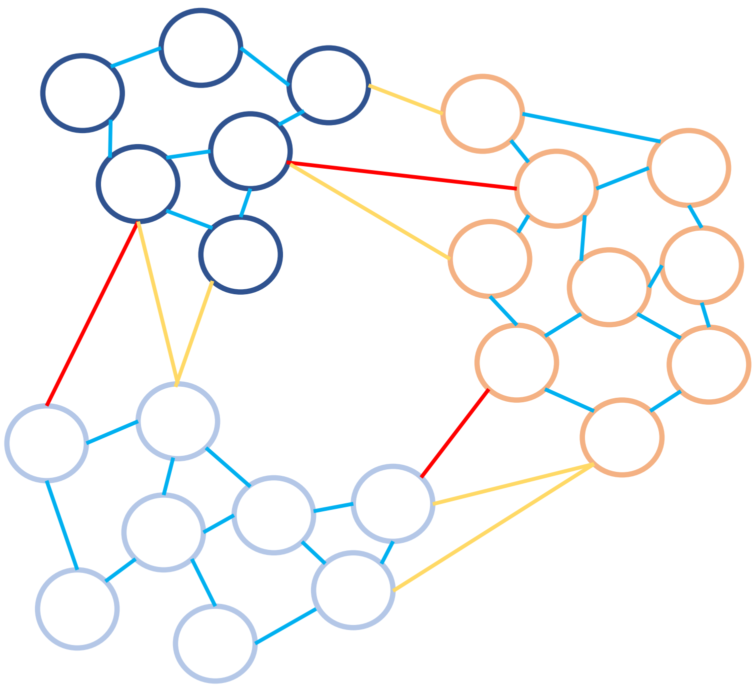


Fig. 4.10 Find the high betweenness edge (red edge is high betweenness, yellow edge is medium, blue is lowest)

The computation complexity is $O(mn^2)$, m is the number of edges and n is vertices. If we want to reduce the computation complexity, one potential solution is to calculate the overall betweenness once and remove them in descending order. However, by our pre-testing, the same betweenness from middle-level edges could profoundly affect the precise classification of vertices, but it is still a way for quick classification. Therefore, in a general way, many paths will be cut off due to the removing process, and they transfer to the next edge nearby. This process could guarantee that the removed edge is the current "hot" edge that most of the communities rely on.

Besides, we carefully consider the potential passengers may be out of the specific social network, and make this social network classification model to avoid this problem. We classify users from a location-based social network (which is more reasonable for users in one city) into several communities and measure the relationship between communities, instead of an individual's relationship. This fuzzy pressing could help the system understand users' social priority in communities, and avoid using the unquantifiable social relationship to indicate the interaction priority in order. Furthermore, different users may belong to different social networks and entirely has no connections. When designing the arrangement algorithm (introduced later), we can consider a three-level priority passenger combination,

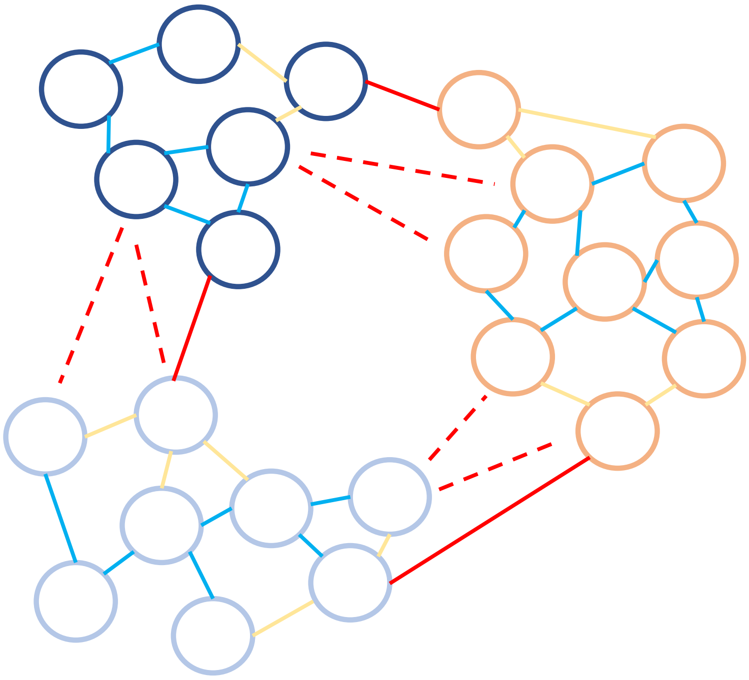


Fig. 4.11 Move them and recalculate the shortest path (red edge is high betweenness, yellow edge is medium, blue is lowest)

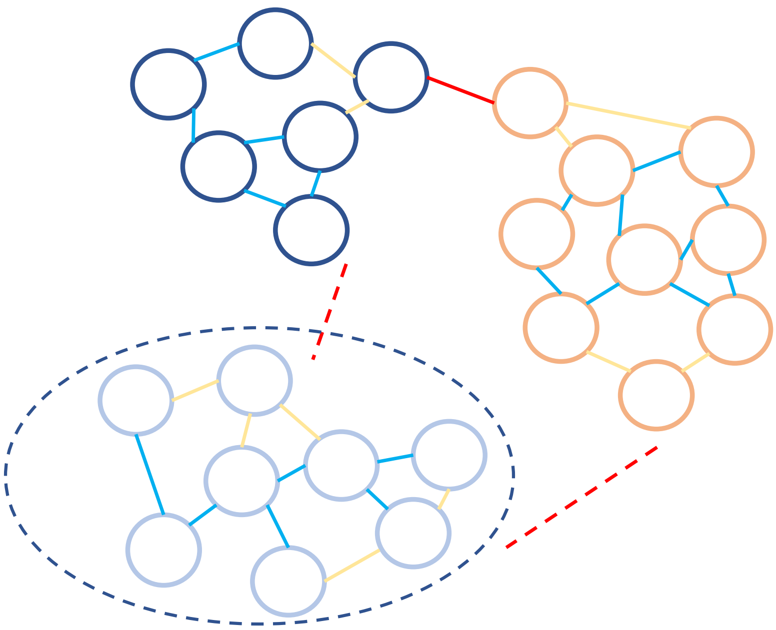


Fig. 4.12 Separate the community if no outside links with other communities (red edge is high betweenness, yellow edge is medium, blue is lowest)

same community, same social network and different network, which covers most of the people in a city and solve the mentioned problem perfectly.

4.2.5 Problem Statement

In this study, given a series of taxi sharing request, our design is to find an optimized route which concludes social network friends in close communities as many as possible, and make sure the travel cost is also minimized among these friends. In a more formal way: Given a set of taxis M on the nodes and edges composed map, and a series of sharing request queue Q in time series, our goal is to design a system that serves this request queue by dispatching idle taxis in M which could satisfy each request in Q, and combine them to improve the QoS with minimum travel distance increase.

Here are two objects for the design: find social network companions in the trip; save money as many as possible. The first object is our primary object, which is to improve the QoS on the journey. Since previous works have already developed various algorithms to approach the maximum economic benefit, it is not difficult to form the maximum economic problem using a greedy algorithm. The real challenge is to search for the pleasant passengers in the same taxi with the relatively small time interval. Generally, passengers all wish to start their trip immediately when they upload their travel plan, and they will abandon this service if time ran out. Therefore, our design could not wait for the next request to form a long queue with maximum span. Instead, we need to design a solution that could either take cost efficiency, fairness, or people-oriented designing, to improve the QoS of taxi service in an all-around way.

4.3 Social Network based Scheduling

The calculation complexity increases exponentially if we calculate all the combination in the queue, which is also a knapsack problem. In order to find the optimal solution, we divide the passenger allocation problem into three steps:

(1) Build a planning map for rough sharing recommendation, and develop a bidirectional taxi searching algorithm to match the path the first passenger demanded.

(2) Discuss a solution that satisfies all the constraints for passengers and taxis.

(3) Develop a social network based algorithm to search for the best solution dynamically until the end of the whole time frame.

4.3.1 Bidirectional Taxi Searching Algorithm

First, after a travel request uploaded, the first thing necessary is to find an available taxi nearby. For the description, we describe the situation using a block map before the extension. The extended blocks only expand the potential set of taxis, which would not affect the core concept of algorithm introduced below. Assume there is a new request q_0 updated into the system queue Q, and the original start point is $Q_0.O$, which belongs to the block B_0. Then, we select the travel time list from the index of B_0, and it lists all the blocks $B_j \in B$ from the nearest to the farthest. Then we assume the current time is t, and for any block j, we have the time constraint here,

$$t + t_{j0} \leq Q.st.l \tag{4.7}$$

$Q.st.l$ is the time limitation for departure, which means the time cost for taxis from other blocks reaching the original should not exceed the user's time limitation. Therefore, any block satisfies this equation becomes a candidate block that may provide idle taxis (or partly empty seat taxis). Since we do have the dynamic travel time list corresponding to each block's index, we can quickly find all the t_{j0} by the descending order, and stop until one block is too far away that holds long enough t_{j0} which cannot satisfy Eq.4.7. Next, we can search for the taxis in these blocks as a set of candidate taxis.

Fig.4.13, the original point in block B_0 can find other taxis in B_2, B_3, B_5. Then each block maps out several taxis in it, which contains idle taxis or rushing taxis that is planned to arrive in a concise time in the future. For those taxis arriving in future, we have

$$Taxi_m.t_j + t + t_{j0} \leq Q.st.l \tag{4.8}$$

where the total sum of taxi arriving, block moving and current time should not exceed the user's waiting limitation.

Generally, the taxis within a limited range should be considered firstly since travel time refers to the travel distance. However, the unpredictable traffic condition may affect the precision of real available taxis, and then we use the periodically updated travel time to improve the error-tolerant rate. More than that, it is impossible to draw all the taxis' future tracks nearby to realize ride-sharing, where searching for all the points in tracks within the travel distance limitation needs higher computation complexity.

After we find the available taxis near the original point, next we discuss the taxis whether the destination is near to the blocks in its future plan. The same with the previous figure, Fig.4.14 shows a bidirectional taxi searching algorithm, in which we search for the blocks within travel time by both sides.

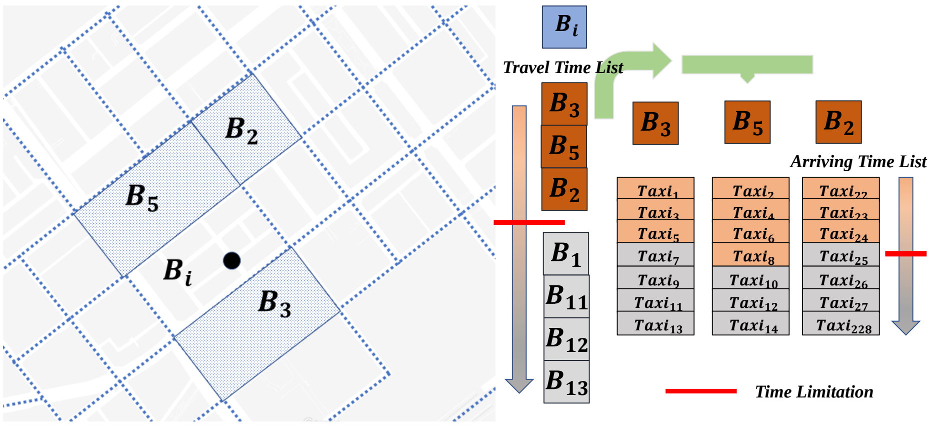


Fig. 4.13 Searching for the nearby taxis from start point

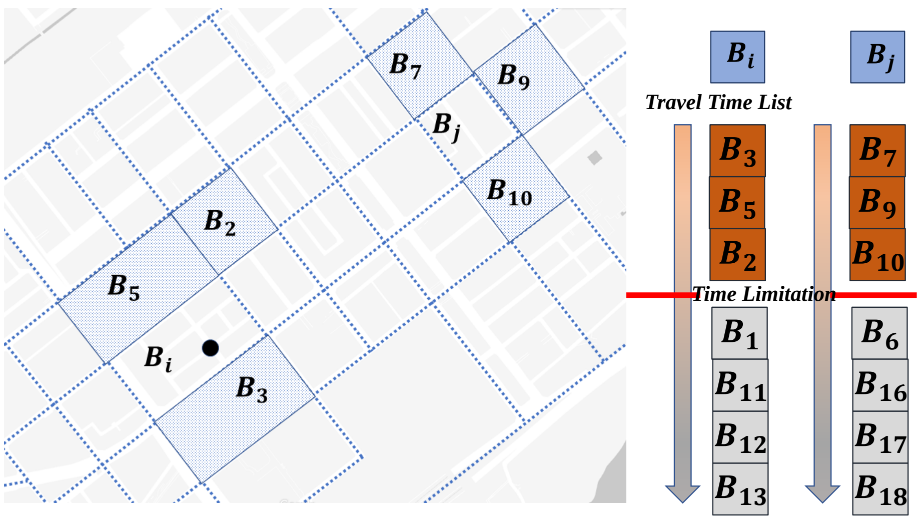


Fig. 4.14 Searching for the nearby taxis from start side and destination side

Let us define the original point and destination point as B_i and B_j, and other high-lighted blocks are the nearby blocks that satisfy the time limitation. Therefore, there are two limitations should be satisfied for the non-empty taxi m passing by the original point B_i and destination point B_j respectively, which are

$$t + t_{xi} + Taxi_m.t_x \leq Q.st.l \qquad t + t_{yj} + Taxi_m.t_y \leq Q.dt.l \tag{4.9}$$

where the taxi should arrive the original point and destination point within the time limitation window $Q.st.l$ and $Q.dt.l$. In this example, the blocks B_2, B_3,B_5 nearby origin, blocks B_7,B_9,B_{10} nearby destination, are selected as the blocks that contain potential taxis for sharing.

Then, we compare the taxis indexed in these blocks to find the intersection on both sides. As shown in Fig.4.15, for every picking and dropping, we list the blocks nearby, from the nearest to farthest. Then, we combine these blocks in each picking and dropping slot, to find the intersection set of taxis from these blocks. For example, the system combines the block B_2 and B_7 at the front of each picking&dropping case, to find the intersection by the ascending order. If it is empty, we pick one of the next blocks in order to continue this process, until we find more than one taxi in the intersection. Here, when selecting the next block, we choose the block nearest to one of the current combinations, in order to make sure the increased distance is minimum.

Through this bidirectional taxi sharing algorithm, the overall travel distance may increase a little comparing with the optimal solution, due to the heuristic indexing method. However, since the index has filtered out unsatisfactory travel distance and time limitation, the taxi searching optimization problem becomes a low computation complexity problem which finds the intersection in a few numbers of sets. Through the process, about half of the taxis are filtered out when almost less travel distance is increased due to the real path selection on the map.

4.3.2 Temporary Passenger Detection and Adding Algorithm

Through the bidirectional taxi searching algorithm, the system can help passengers find suitable taxis, for a single journey or sharing. The passenger can use an idle taxi nearby enough to establish a new schedule. However, for the case of sharing, we never discuss the influence on the passengers who are already sitting in a taxi.

Here, we use a taxi status matrix M to indicate the current necessary information and travel plan, where the estimated arriving time is indexed in each block. The main purpose of the system is to satisfy the user request queue Q as many as possible. Therefore, when

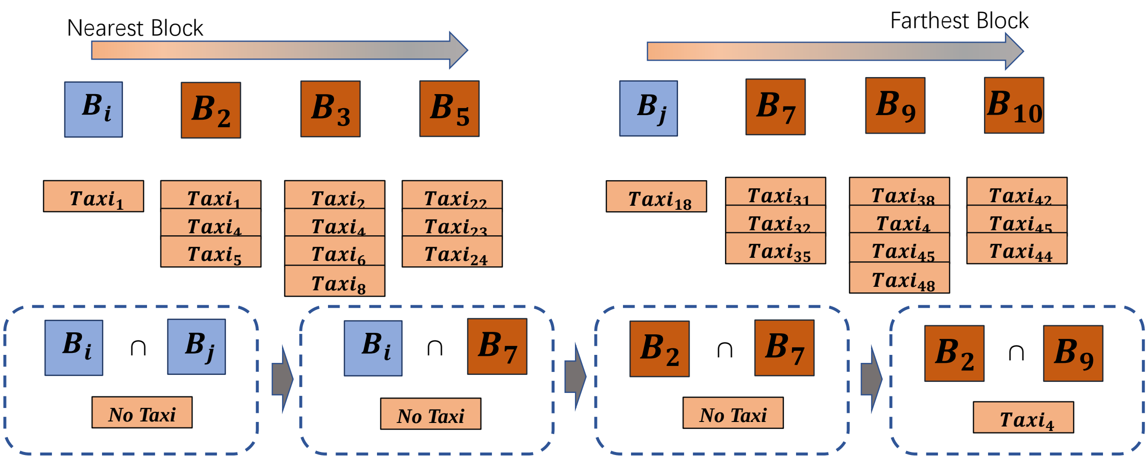


Fig. 4.15 Searching for the taxi through block intersection

the single schedule for the first user is established, the next things need to be done is to calculate whether those potential members are suitable for sharing. The new intersection could be found in the next two ways: new picking and dropping point are inserted in the current schedule; The path of the current schedule is rerecorded at once. After finding one intersection, we estimate whether the constraints can be satisfied. There are three steps should be concerned before new riding: The capacity is still enough for the new rider; Time and economic cost constraint could hold; Since a new set of passengers is gathered, we reorder the path to guarantee the travel distance in minimum.

It is easy to confirm the current capacity status by the system. Then, we discuss the time constraint that all the current passengers and new riders should satisfy the time limitation. Fig.4.16 shows an example that the estimation for a new intersection appears. The current path is from the picking up point $Q_1.o$ to the dropping point $Q_2.d$, and this taxi passes $Q_2.o$ and $Q_1.d$ as well. Meanwhile, a potential rider n_3 is inserted into this schedule (noted as dash line), and we discuss whether the constraints can be satisfied. To achieve the travel distance optimally, the system suggests the taxi go to $Q_3.o$ firstly after picking up user $n = 1$ at $Q_1.o$. Then, the taxi picks up n_2 at last and drop them one by one. Notice that the optimal path relies on the real traffic condition, the path for real travel time in optimal may be different from travel distance in optimal. Therefore, the optimal path should be calculated once more after the new schedule is established. Without loss of generality, in the beginning, we use the shortest distance to arrange an optimized route since the calculation for multiple paths is a significant burden for the overall schedule optimization.

Next, we formulate the time constraint for the new ride-sharing schedule. We use $T|O_x.o \to O_y.o|$ to indicate travel time of current path from point $O_x.o$ to point $O_y.o$, and

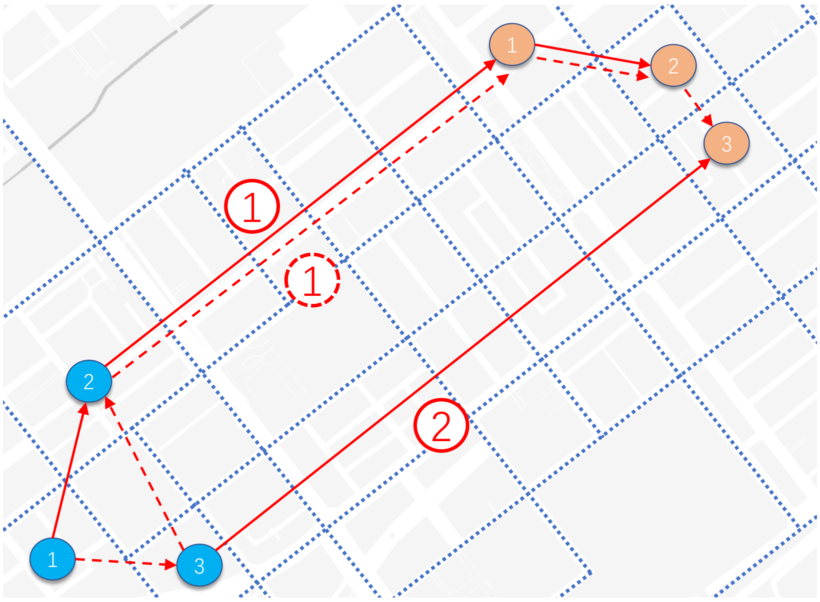


Fig. 4.16 An example that adding new passenger to the exiting travel plan

tw_x is the time wasted at point O_x. Then, the increased pick up time and arriving time tds and tdd for this taxi noted as,

$$tds = T|Q_1.o \to Q_3.o| + tw_3 + T|Q_3.o \to Q_2.o| - T|O_1.o \to O_2.o| \tag{4.10}$$

$$tdd = T|Q_1.o \to Q_3.o| + tw_3 + T|Q_3.o \to Q_2.o| - T|O_1.o \to O_2.d| \tag{4.11}$$

Here, we only assume the situation that the taxi stops and is waiting at $O_3.o$ for the new rider. From the equation, we can know the reorder for user n_3 may result in the increment of arriving time for former user n_1 and n_2. Thus, for user n_2, the picking and dropping time limitation should be satisfied as,

$$t + T|O_1.o \to O_2.o| + tds \leq Q_2.st.l \tag{4.12}$$

$$t + T|O_1.o \to O_2.d| + tdd \leq Q_2.dt.l \tag{4.13}$$

where the increased travel for new rider should not exceed the time limitation. The above process is to filter out whether the time constraint can be satisfied for a specific new rider and find the optimal solution. Next, we discuss how to filter out and select the potential candidate for sharing through social network communities.

4.3.3 Social Network based Heuristic Algorithm

Require: Set of candidate users under a cover N, idle taxis M' , user request queue Q;
Ensure: Optimized Schedule S with picked passengers m;

while $q \leftarrow 1$, $q <= size(M')$ **do**
  Establish a path $Path_{Q_1}$ for Q_1 ;
  Find idle taxi M'_q ;
  for $j \leftarrow 1$, $j <= a_m - 1$ **do**
    Find potential riders $Q' = MatchPath(Q, Path_{Q_j})$;
    Recognizing Community Members in Q' ;
    $MemberExist \leftarrow 1$;
    while $level \leftarrow 0$, $MemberExist == 1$ **do**
      if $find(Community(Q') == level)$;
      Pick up rider set R from 'Q' in this $level$;
      $MemberExist == 1$;
      end ;
    end while
    Find the optimal schedule with one passenger y in R;
    Add passenger y into idle taxi M'_q;
    Establish a path $Path_{Q_j}$ after adding one passenger in R;
    Delete passenger y in the Q;
  end for
  Return Travel Plan S_q;
end while
return S ;

Algorithm Objective:

By map planning, we divide the whole city map into blocks, and all the useless information can be filtered out efficiently. Given a subproblem that one passenger is scheduled in a taxi and available for sharing, we know how to add potential riders and minimize the travel time and cost. In another word, the system can search for the optimized solution in a limited sized subset. However, the overall searching process increases exponentially by the number of n' user request in the queue, and the computation complexity is within $O(n'^{a_m})$, where a_m is the capacity for the specific taxi m. We need to develop an algorithm that solves this subproblem within several time slots to reply to the dynamic queue [22]. The main idea of the algorithm proposed here is finding the passengers who have the most significant influence to save the cost and improve the QoS. At first, we select the user at the head of the queue to start a new schedule (first come first served). Then, we run a parallel path-finding

Fig. 4.17 Double path searching on the real map and social network map, to find the intersection of close passengers based on social relationship and real travel plan

algorithm to search for the potential passenger set near to the original path, through the real map and social network community map. Finally, we compare all the options from the multiple paths created by the given set of passengers, and select the best one. Following is the detail explanation of our proposed heuristic algorithm.

Problem Reduction: This problem is to search for the best solution about, where one of the main purposes is to find the highest economic benefit. For calculating the integrated economic benefit in one taxi (or one subproblem), we have

$$EB = \sum EB_j \qquad \textbf{where} \qquad j \in M \tag{4.14}$$

By observation, we find saved economic cost EB_j is more likely influenced by farther path. Meanwhile, we try to put passengers in the same community as many as possible, to increase the service experience of ride sharing. Therefore, in one taxi, it is an alternative selection that we consider the passengers in the same group first, which may sacrifice part of the economic benefit. Based on the principle of first come first served, the system picks up a passenger at the head of the queue, to establish the first path. Then, it takes the potential passengers in the same community, and passenger save more money would get the priority. Notice that it is a fuzzy conception about the relationship strength for users in the same community, and it is hard to give a strict standard that QoS of one combination is undoubtedly higher. Instead, we find the combination with higher economic benefit is much more perversive, which is also the main purpose of the activity "sharing" itself. Thus, in the same community, we only consider the member who may save more cost.

The above analysis is for the candidates in one community are more significant than the capacity in one taxi. Although in location-based social networks, it is possible that members in the same community appear in the same local area since they have the same social features and activities. However, in the more general situation as shown in Fig.4.17, the system still needs to find potential passengers out of the community to fulfill the capacity and save travel cost. Therefore, the system searches for passengers in the nearby community (one hop community) for farther path passengers. Then, on the social network map, the system emissions to find close passengers to increase QoS, while the new adding of passengers shrink the options of the travel path.

Algorithm Description: Based on the above theory, we have the following scheduling steps for passenger allocation:

(1) Input the user request queue Q and select the first user at the head of it to establish a path. (2) Select all potential passengers from the queue, which satisfy the primary constraints, noted as Q'. (3) Pick up users who are both in the queue Q' and the same community of the first user. If more than one is selected, pick the passenger could save the money most. Then, reorder the schedule to optimize the travel route. Otherwise, skip to step 4. (4)With the fixed passengers in one taxi, filter out users in request queue Q' again. Pick up users who are in the same community and renew the queue. If no one is satisfied, find passengers within one hop in a social community map, then pick up the user needs the farthest path on the real map. Then, renew the schedule to shrink the temporal request queue further Q'. (5) Repeat the step 3 and 4 until the capacity of one taxi is fulfilled. If no one could be found in the queue before fulfilling, stop the process for current taxi and start step 1 using another taxi.

Algorithm I show the pseudo code of this heuristic algorithm. Notice that this algorithm is generally applied to the situation $a_m \leq 4$. The algorithm may not perform well if the capacity of the taxi is more than 4, due to the size limitation of community, which makes fewer users in one community appear in the same time. However, the algorithm considers the user experiences as the priority, so the user request at the head is fulfilled first and most of the request will not be ignored. Besides, we also provide 'comfortable' companions to reduce the travel cost, which strives to make the journey more pleasant and economical.

Computation Complexity: The algorithm will terminate if possible combinations using a different number of passengers are investigated. We preset i_{max} as the number of taxis online, and assume $i_{max} >> w$ and j_{max}. The constraint examination takes less than $O(i_{max}^2)$ to establish Q' (pick each candidate $i \in N$ and estimate the remaining $i_{max} - 1$ passengers). The circulation related to capacity a_m is normally a constant loop while the two options inside have the complexity of $O(i_{max})$ for each of them. Since they belong to the same nested loop, the overtime complexity is less than $O(i_{max}^2)$.

Dynamic Social Network Based Scheduling

Algorithm 2 is the pseudo-code of a dynamic scheduling algorithm in order to solve the problem dynamically. In each time slot, as introduced in the part of the framework, we input the temporary candidate users N' and taxis M' into the system. Then, we use algorithm 1 to calculate the result in each time slot and algorithm 2 to update new request and traffic information from the map. The above procedures repeat and terminate until the end of the time frame t_{max}.

Dynamic Social Network Based Scheduling Algorithm;
$t = 0$;
while $t \leq t_{max}$ **do**
  Input candidate users info N', available taxis and travel plan info matrix M', current request queue Q and economic benefit EB;
  $Q = NewRequest(Q)$;
  Run $[Q_t, N_t, M_t, EB_t] = Algorithm1(N', M', Q, EB))$;
  Remove users scheduled in Q_t ;
  $t = t + \Delta t$;
end while
Return $Q = Q + Q_t$;

4.4 Performance Evaluation

First, we made a prototypical application based on the structure and algorithm we proposed before, which contains mobile devices as a client and a standard PC as a server. The real implement test run by mobile devices and the large-scale numerical analysis run by the PC server, which is introduced below.

Application Test: We run the test on our application using mobile devices. In Fig.4.18, we start from the front of a university and select the destination near the Bokoi railway station through the interface, which is a famous railway station but no direct path of public transport medium. Therefore, sharing a taxi is an economical way to reach the destination. We submit our basic personal data to the system within the available period from 10:50 to 11:10. Then, the system accepts our demand and replies to us with a detailed result of

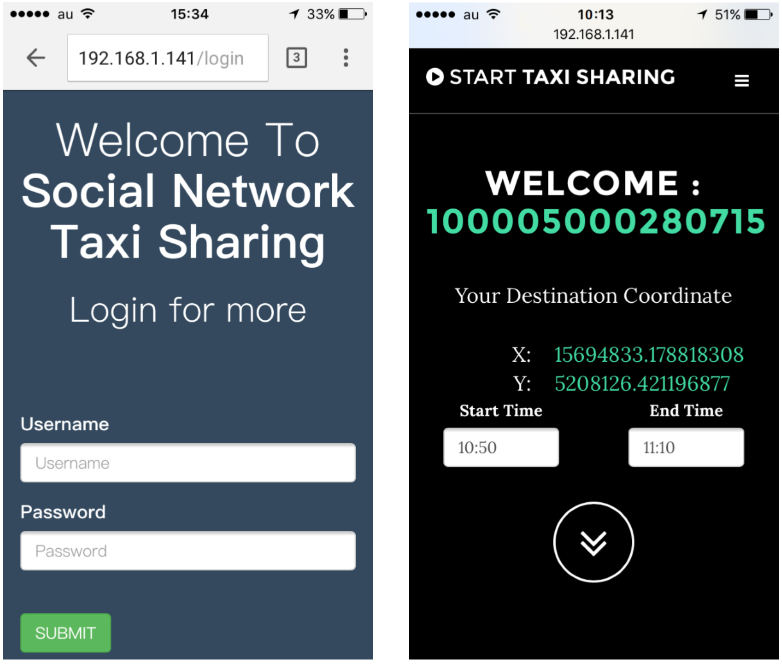


Fig. 4.18 Screen shots of an experiment in designed application (A)

the sharing arrangement. A demonstration video is shown on our website (http://www3.muroran-it.ac.jp/enes/%7Ecfzhang/researches.html).

Comparison Algorithm: Besides, we simulate the proposed social network based taxi sharing service using the proposed algorithm (noted as SONETS) described in section 5. We perform the simulation in the scenario of Muroran City. Besides, we choose the other three algorithms for comparing whether the proposed system can accomplish the goal. One of the social network based scheduling algorithm only taking users in the same community into consideration,(noted as SONETS-Single,SS). Another is a similar algorithm which considers nearby communities, (noted as SONETS-Near,SN). The last one is an optimized distance algorithm that only concerns the distance optimal to achieve better economic benefit (noted as Cost-Optimized,CO). It runs only the part of economy benefit constraint and without any concerns about the social community priority. Notice that the CO is a brute-force algorithm that calculates all the possible path when the first passenger's plan is established, which is the optimum algorithm within the complexity time $O(i_{max}^2)$. The calculation time is $O(i_{max}^4)$ for the real optimum algorithm to calculate different goals. However, since the first travel plan has constrained the potential passenger set, this algorithm runs in a relatively small case. Various standards are also recorded to compare the performance of these algorithms.

Practical Experiment: Firstly, we dig the local friend dataset on Facebook as our potential consumers, who are all students and faculties in Muroran Institute of Technology.

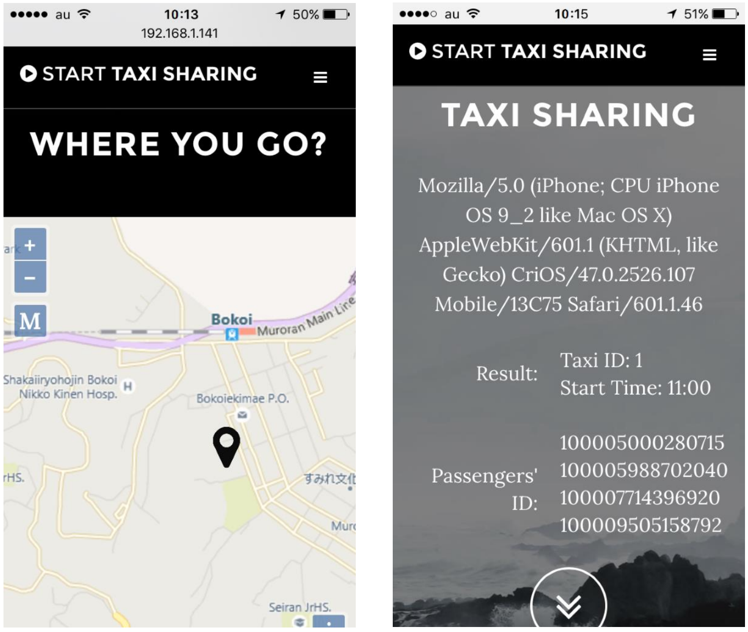


Fig. 4.19 Screen shots of an experiment in designed application (B)

The number of our potential users is more than 350 and all of them have a direct or indirect relationship with the others. We find these users could be divided into communities automatically, which separated based on the different research areas. Due to the size limitation of social network dataset, we extend the experiment by inputting location-based online social networks from Stanford large network dataset collection. The set of Brightkite location-based online social network is analyzed here, where more than 214000 edges are considered as our potential users and match the size of our city. The destinations of these passengers are distributed on the city map blocks, based on an adjustable request ratio. Notice that the real probability of a request can be extracted from the trace studies or other online taxi services from previous works.

We find the peak time empirically for taxi sharing every day is about three hours, which is considered as the length of the taxi sharing service window for the simulation. During this period our potential consumers wait at each block, which considered as the start point. The available time window for each request (available slots, requesting time, etc.) is randomly picked in our simulations, following the uniform distribution with a fixed average length. Points at the edge of each block are considered as the start and destination for users.

Analysis of SONETS: Next, we test the setting of parameters in our proposed system. The price of all the taxi cost (e.g., P_0, P_1) for consumers is based on the price of daytime in Hokkaido, Japan. We consider the time frame T is two hours. The taxi distance limitation Z_0 and Z_{max} are set as 2 km and 10 km. The lower limit Z_0 is the minimum charge distance, and

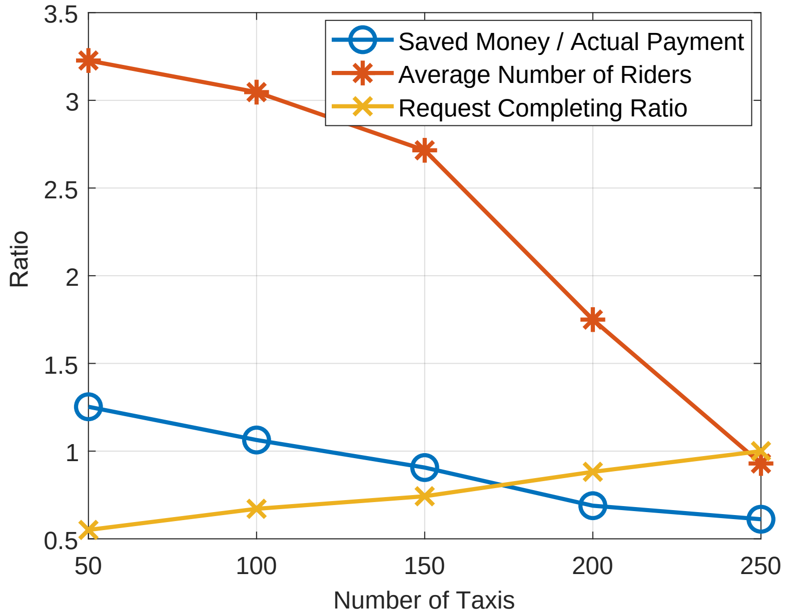


Fig. 4.20 Different measurements changed by number of taxis

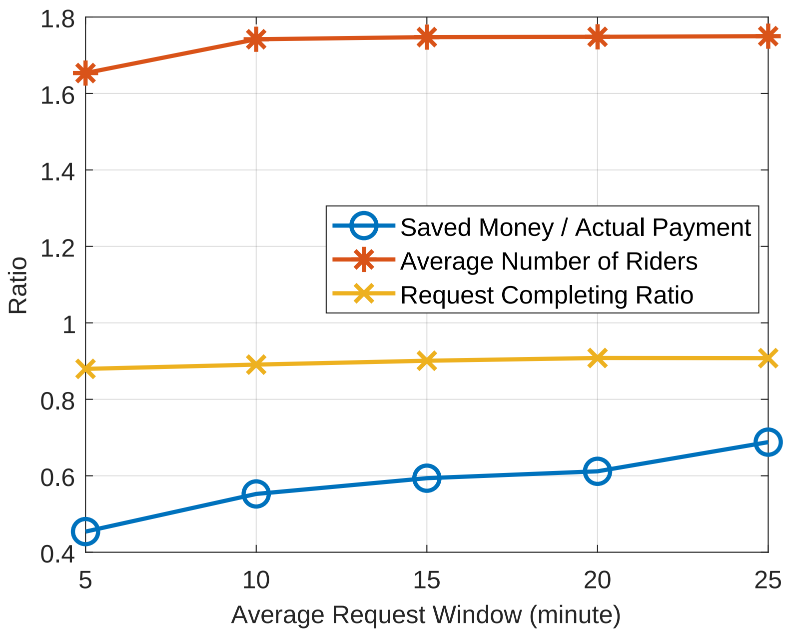


Fig. 4.21 Different measurements changed by different time window

using bicycles or walking is cheaper and flexible for a shorter distance. The superior limit could almost reach the edge of our city, and use the public transportation system is more economical when the travel distance is above 10 km. We divide the map into about 10000 blocks, based on the method in Section 4. Then, two parameters need further discussion, the adjustable average time window of each request and the taxi density in the city.

In Fig.4.20, three measurements are discussed here. The first is the ratio of Saved Money / Actual Payment. As mentioned above, the saved money is calculated by Eq.4.6. The actual payment is the sum of real travel cost for all passengers. The average number of riders is the expected number of riders in a taxi. The request completing ratio is the ratio that the requests from users are fully satisfied before time runs out. Many situations may lead to the skip of taxi request, e.g., no taxi nearby, no idle taxis, no bonus for drivers. Here, we conduct the situation that drivers are willing to serve for these passengers, since more travel distance means more revenue, and this is in line with personal will.

The x-axis is the number of taxis in the city, randomly started in blocks. The real condition is that in usual time more than 50 taxis are in the city, and we set the request ratio as 0.8% in each block. The ratio of request completing increases with the number of taxis. At first, only 55.07% requests can be responded, due to the lack of idle taxis. However, with the increase in taxis, more and more requests are responded. The ratio of Saved Money / Actual Payment decreases by the number of taxis. This is because the overflow of use requests provide more options for taxis to maximize the revenue and choose a relatively longer schedule. However, the ratio decreases to 61.16% due to the smaller size of the request queue, since the taxis can serve all the requests at once when a new request is updated. The ratio of the average number of riders decreases due to the increment of taxis. At first, almost all the taxis are occupied by more than three users, where the system can make more travel plans to maximized the travel cost and picked up passengers in a close community. Finally, the occupancy rate falls to 1, which means part of the taxis remain idle.

In Fig.4.21, the same measurements are discussed here. The differences are, we adjust the average request time window instead of the density of taxis. Generally, it is acceptable that users wait for taxi coming within 5-25 minutes, which depends on the personal habits. From the figure, we can observe that the ratio of Saved Money / Actual Payment increases significantly with the increase of time window. Since longer waiting time means more options in the request queue, the system can arrange travel plan more efficiently to save more money for passengers. Finally, it reaches almost 66.8% if the acceptable waiting time is within 25 minutes. The request completing ratio increases rarely. That is because the number of taxis can cover most of the request queue, which keeps the ratio at about 90%. The average number of riders shows that still part of the passengers take participate in the sharing activity (idle taxis are also calculated). The request of a time window within 10 minutes shows the same result with the longer one, which means all the requests can be well responded before skipping. The time window of 5 minutes is too short for idle taxis arriving at the rendezvous.

Performance Comparison: Next, we discuss the performance using different algorithms. As mentioned above, the SONETS-Single and SONETS-Near are the algorithms only consider sharing partners from nearby communities. Notice that, using the location-based social network dataset, we divide the almost 10000 nodes into 200 communities. Then, the last outside links connecting with other communities are recovered, and these specific communities are considered as the single hop communities. The community within two hops which are also considered as 'close' communities for SONETS. The purpose of comparing these social network algorithms is that we can analyze how much economic benefit should be sacrificed to achieve higher QoS trips. It is also a tradeoff between system service capacity (overall taxi capacity) and higher service requirements (sharing wiliness with outside communities).

Fig.4.22 shows the overall saved money within three hours. We set about 200 cars on the city map, and the request ratio in each block changes from 0.2% to 1%. Without question, the distance optimal algorithm approaches the best performance about economic benefit, and remains at the same standard when the request ratio is higher than 0.4%. It is because all the taxis have already been fully loaded. The same trend with the distance optimal algorithm, our proposed SONETS has no significant difference, and approximate to the above algorithm when request ratio increases. It is because more potential passengers are in close communities, and optimize the travel plan could get a higher economic benefit. The SONET-single algorithm only considers potential passengers in the same community. Therefore, it is difficult to find partners until the request ratio becomes almost impractical. The left one, SONET-Near algorithm considers almost 1/3 members in the queue, which could perform the same trend with former algorithms.

Fig.4.23 shows the sharing efficiency, which uses the ratio of Saved Money / Actual Payment to estimate the necessity of sharing. The same results with the former figure, four algorithms increase with the request ratio. The same with a received view, the sharing efficiency is unsatisfied if the request ratio is low enough. Based on the results, it is no specific meaning for sharing if the ratio is lower than 0.4%, since another cost such as increased travel time has not be taken considered yet.

Fig.4.24, it shows the average number of riders in one taxi. At the start, when the request ratio is low enough, it is difficult to find sharing partners for all the four algorithms. However, when more request appears, the distance optimal algorithm serves as many passengers as possible, to achieve a higher seat ratio. Our SONETS consider community members first,

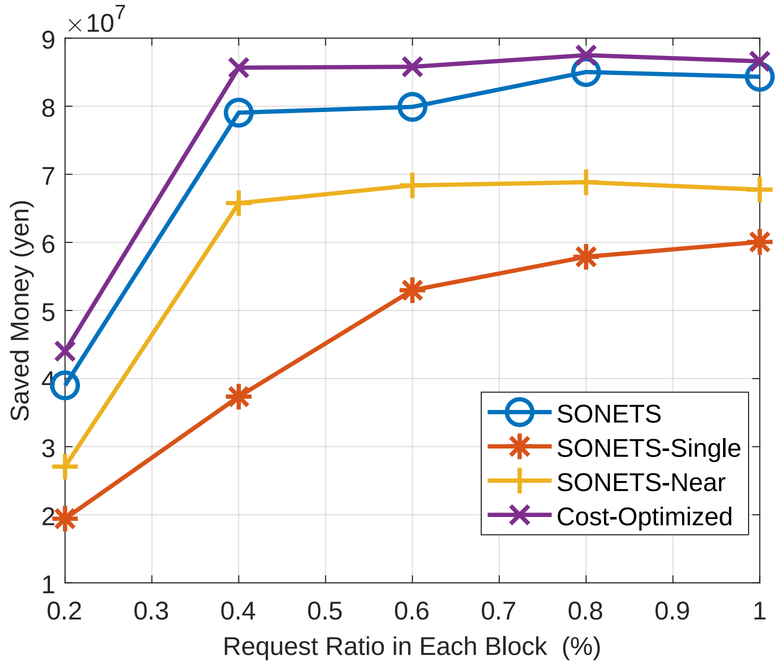


Fig. 4.22 Saved cost (economic benefit) Comparison

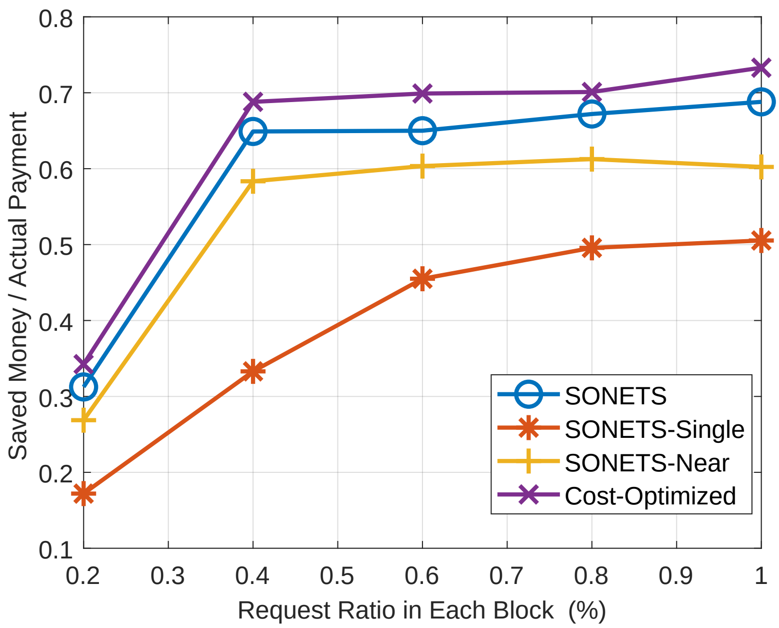


Fig. 4.23 (Saved cost / Actual cost) Expense Efficiency Comparison

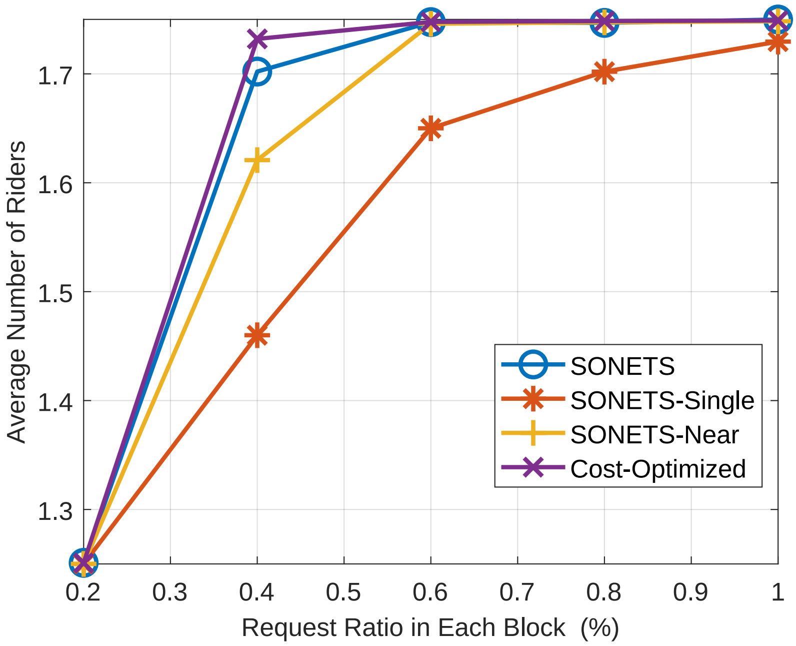
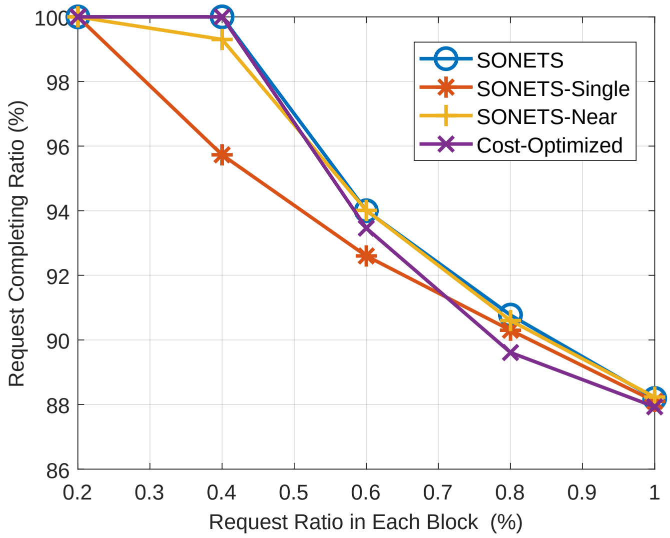


Fig. 4.24 Average number of riders

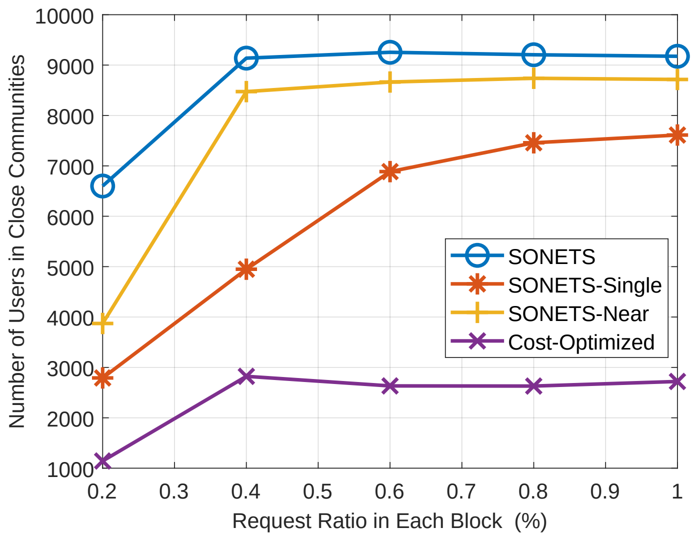
leading to fewer options than distance optimal. However, it still approximates the former one since other passengers are also picked if the seat is empty. The seat ratio of SONETS-Single and SONETS-Near is lower due to more strict standard for partners. However, they all get an acceptable seat ratio when request ratio increases.

Fig.4.25a is the request completing ratio changed by request ratio in each block. At first, due to the lack of requests, all the algorithms could handle the new request updated and serve them immediately. Our SONETS performs better in all around ways due to the 'first come, first served' principle, and less utilitarian to pick up community members that may save less money. The higher density of community members in the queue, the higher possibility to combines them, which releases the opportunity for other communities for the next schedule. The distance optimal (notes as a cost-optimized algorithm) trends to consider highest profit passengers, when lower profit passengers are ignored all the time, leading to the lower completing ratio. The other two algorithms achieve lower completing ratio since some passengers outside the community does not take into consideration, which leaves some empty seats and makes the system not fully loaded.

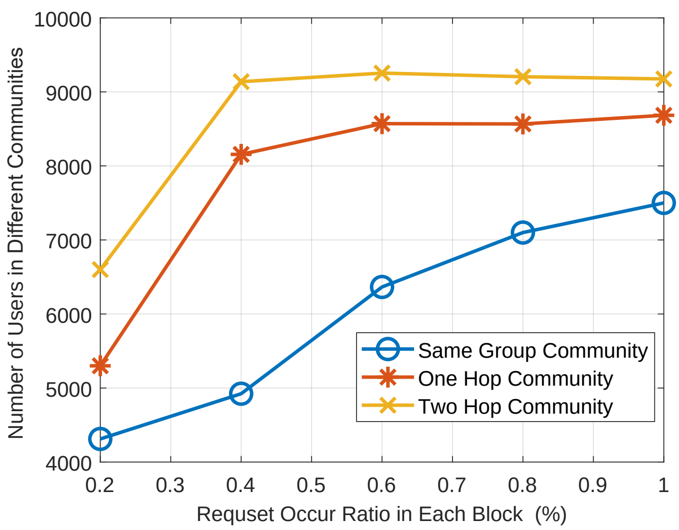
Fig.4.25b is the statistical result of 'close' companions considered in the process. Our proposed SONETS holds the highest record, due to the greater coverage in social network communities. The same with SONET-near, they reach the bottleneck to explore new community members because of the system capacity. The algorithm only considers the same community also approaches the limitation when the request ratio increases to 0.6%. The



(a) Request completing ratio changed by request ration



(b) Statistical result of ’close’ companions considered using different algorithms



(c) Statistical result of ’close’ companions

Fig. 4.25 Performance about close community members cost-optimized algorithm remains at the relatively lower standard for picking their same community members, which shows the same result with randomly picking. Never than less, we also count the number of passengers in different communities in Fig.4.25c, to analyze the composing proportion of passengers. At first, due to the lack of request, SONETS could not find enough members in the same community, and have to explore outside communities to achieve better sharing efficiency. However, with the increase of request ratio, it is possible to search for potential passengers in the nearest community, and continuous increase with the request ratio, the same with SONET-Single algorithm. Therefore, when request overflows, our proposed algorithm still try to improve the sharing environments (as well as QoS), although the system capacity has already become a limitation.

Through the comparison, we show that the cost efficiency, fairness, and people-oriented designing are the three features of our proposed algorithm. Instead, our passengers only need to sacrifice 1.16% cost to experience more pleasant and reassuring sharing service than usual carpooling in rush hours, which is acceptable for most users.

4.5 Conclusion

In this paper, we study the problem of taxi resource allocation that optimizing the trade-off between travel cost and user experience. The new proposed system considers social network communities which helps improve the quality of the sharing service. Through map planning, we concentrate on the city blocks' connection with routs to make an efficient booking service. After that, the passengers within close communities are recommended by the proposed SONETS algorithm. In the experiment, we make an application and apply it well in the real environment. The results of large-scale simulation also show that our algorithm can provide cost efficiency, fairness, and humanized solutions for potential users, which leads to a higher quality of user experience.

Chapter 5

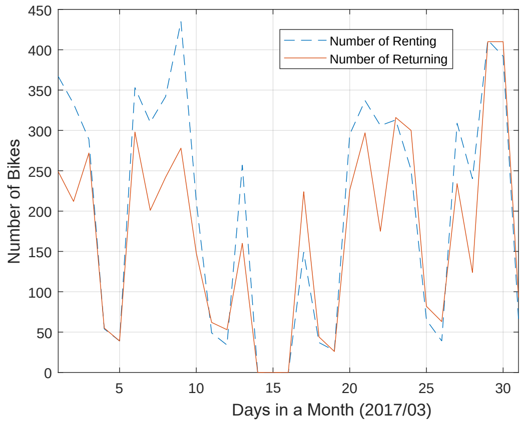
Battery Maintenance of Pedelec Sharing System: Big Data based Usage Prediction and Replenishment Scheduling

This chapter describes the efficiency-oriented maintenance service using AI prediction, to show a comprehensive solution how to deal with the sustainability issues with the development of urban ITS.

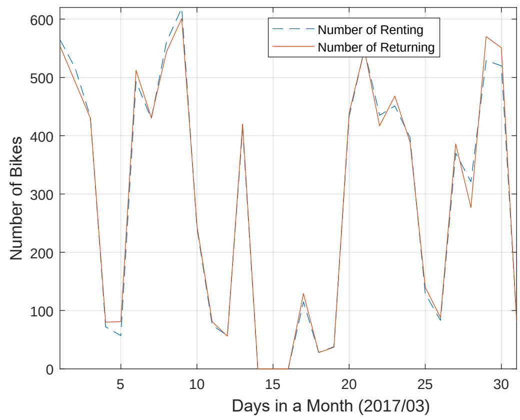
This chapter develops an optimal path planning scheme for replenishment trucks based on big data analytic and prediction. To the best of our knowledge, we are the first to use the systematic methodology of Analysis/Modeling/Test to advance the bike prediction process using the captured datasets, and conduct the hybrid prediction process learns internal connections between different dataset dimension as well as optimize the overall maintenance efficiency.

Our contributions are three-fold: Big data based prediction: we develop an efficient offline prediction scheme based on big data analytics. We use local predictor to analyze the characteristics of the station itself, global predictor to analyze global correlations, and inflection predictor to deal with unexpected situations. An prediction integrator is used at last to improve the prediction accuracy; Optimal path planning algorithm: based on the predicted results, we can analyse the degree of correlation between stations in real time. After making a station correlation map, a new pedelec maintenance algorithm is developed. It is able to avoid the problem of repeated charging and charge nodes with high demand, thus improving charging efficiency; Evaluation and verification: we select NYC's shared bicycle data and city-relevant big data for real simulation experiments. After comparison of prediction accuracy, real-time average power and overall service quantity, it is proved that our solution is superior to most benchmark algorithms.

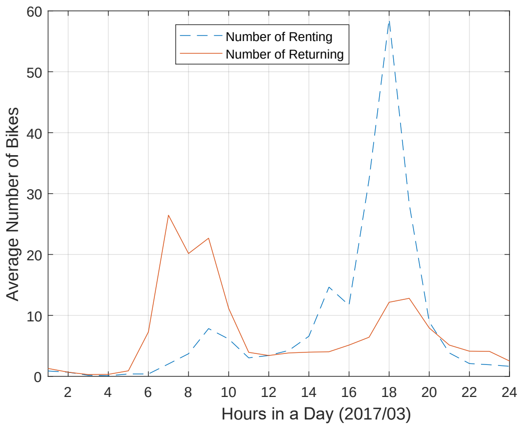
The remaining parts of this paper are structured as follows: Chapter 5.1 discusses the preliminaries of urban big data and then provides a brief overview of our hybrid pedelec system for prediction and maintenance. Based on the process of the proposed framework, Chapter 5.2 discusses the local predictor and global predictor that complete the predictions using temporal and spatial information, respectively. Next, it discusses how the apparent inflection impacts the prediction of each station, and finally, an integrator is used to combine these predictive results. In Chapter 5.3, we develop a weight based algorithm to solve the battery replacing problems of replenishment trucks. Chapter 5.4 conducts a large-scale simulation to test different methods, and Chapter 5.5 closes the paper with concluding remarks.



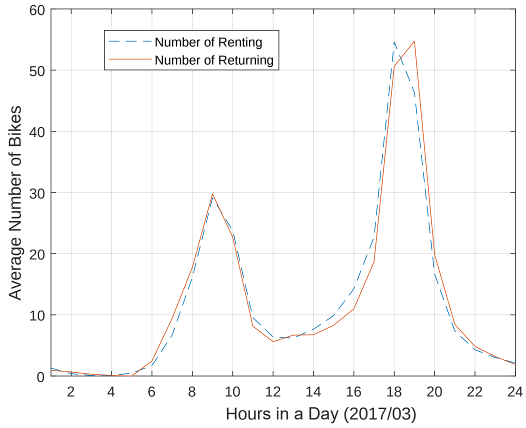
(a) Number of bikes changed by date (station A)



(b) Number of bikes changed by date (station B)



(c) Number of bikes changed by hour (station A)

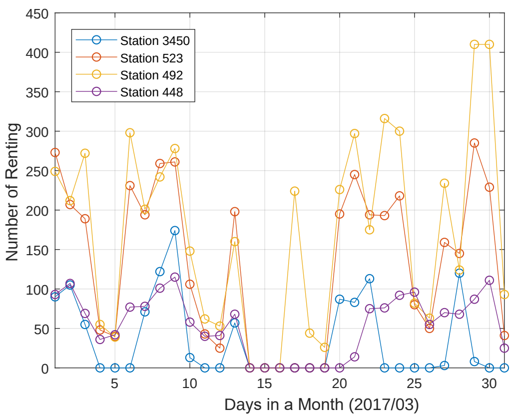


(d) Number of bikes changed by hour (station B)

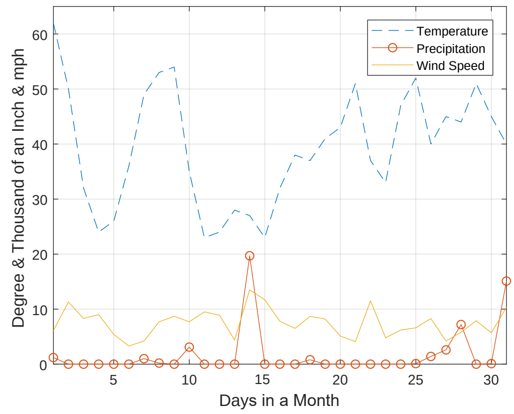
Fig. 5.1 Preliminaries for bike sharing records. The statistical result ranges from different hours and days, for analyzing the potential connections between bike usage and local meteorologies

5.1 System Model and Overview

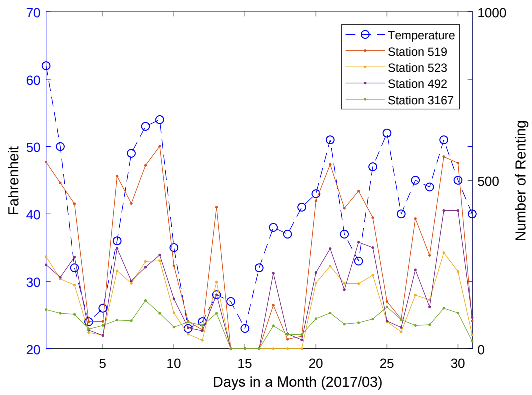
5.1.1 Impacts of Urban Data on Station Usage



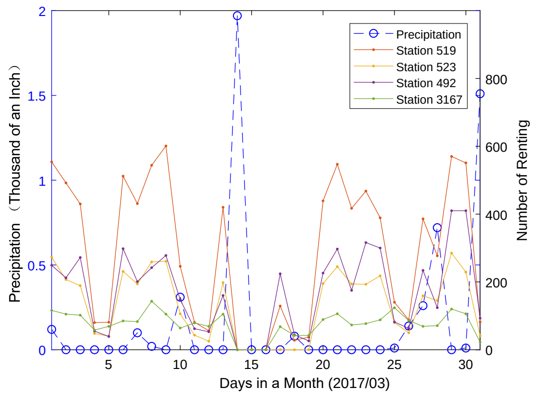
(a) Three nearby stations of A



(b) Weather condition changed by date (2017/03)



(c) Number of bikes influenced by temperature



(d) Number of bikes influenced by precipitation

Fig. 5.2 Preliminaries for bike sharing records. The statistical result ranges from different hours and days, for analyzing the potential connections between bike usage and local meteorologies

We first study the interplay between the big citywide data and the pattern of bike usage which shares insights on the developed prediction scheme. Through our preprocessing of urban big data, it is found that the bicycle usage data have strong correlation with the characteristics of the station itself and the global influence: the station location determines whether the station service type is used for asymmetric travel during busy periods (unbalanced access), central transit station (high throughput) or edge extension node; Weather, large-scale activities or temporal high throughput stations around the station will also affect the use of the stations around an area and cause fluctuations in the use of the whole area.

A bike usage record set X includes time information, station information, customer information, and other decryption information. For a single record x, the duration time is noted as t_x, which is related to the start time $t_{s,x}$ and end time $t_{r,x}$. During the renting period, it forms a trajectory from start station ss_x to return station rs_x. Then we make statistics to build an index for each station, including the usage amount in different hours and dates. Fig. 5.1a and Fig. 5.1b show the usage amount of starting and returning in station A and B, respectively. Note that station B is a popular station at downtown and keeps the highest usage record in March 2017. Station A is the second popular station this month. We can observe that station B is a 'center node' in the station network and reflects the usage trend of the whole city, while station A seems to be a functional area that forms unbalanced demand of bikes. Through Fig. 5.1c and Fig. 5.1d, we find out that station A is in the resident area since the unbalanced usage ratio in the morning and closing time. Meanwhile, station B is a civic activity area after closing time. Combining with the results, we can see the daily traffic flows and detect unusual conditions based on the prediction.

Another high relevant data set is the weather condition, which is similar to [51]. In Fig. 5.2b, it shows the temperature, precipitation and wind speed in March 2017. In Fig. 5.2c and Fig. 5.2d, the usage amount in 3 days (14th-16th day) is relatively low, as well as the temperature. Another observation is that the usage is more sensitive to unusual weather (e.g., rain) since there is nothing to protect cyclists. It may be related to the uncomfortable weather since the low temperature and high precipitation in these days. Fig. 5.2a is the usage of nearby stations compared with station A. The usage from stations nearby follows the same trend of increase and decrease with station A. The stations far away from station A have more or less opposite trends with it in.

In a word, these temporal datasets profoundly affect the usage amount of each station. Meanwhile, the spatial datasets such as coordinators and relative locations reflect the specific features in different areas [5]. Therefore, when building the predictive model, we take these temporal and spatial features into consideration.

5.1.2 Structure of Scheme

Fig. 5.3 shows the structure of proposed pedelec maintenance system, including three processes: (1) Preliminary Data Processing; (2) Prediction Process; (3) Maintenance Service.

Preliminary Data Processing: The process of preliminary data processing mainly handles with raw data from temporal and spatial datasets. It also provides the baseline of the predicting outcomes from local predictor and global predictor [50]. The prediction process distinguishes the inflection situations in which the usage changes more sharply than a regular standard and finally aggregates the above results, and the final process aims to provide

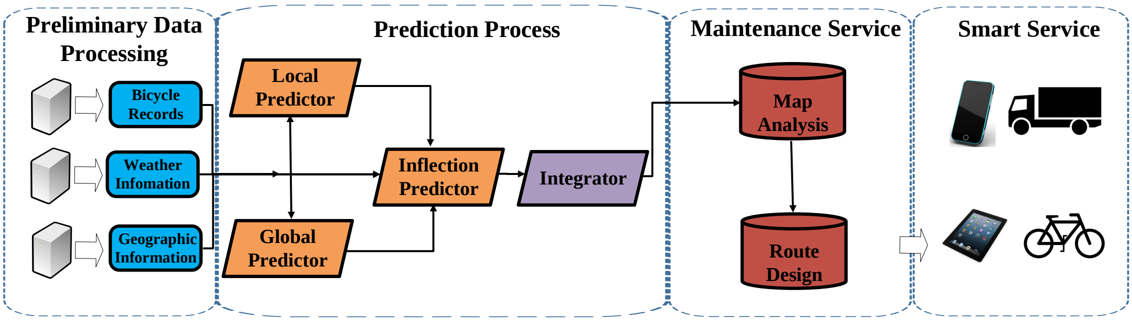


Fig. 5.3 The framework of the pedelec maintenance system. Our work contains three part, preliminary data processing for raw data, prediction process for usage trend prediction, and maintenance service for pedelec maintenance tasks

the routes for each maintenance fleet based on the calculation of our sharing maintenance optimization algorithm.

Prediction Process: Firstly, it loads the raw big data sets through different resources and formats them to meet the predictive model [37]. The raw temporal data includes all bike usage records in recent years, the weather report. The raw spatial data includes station coordinator and local meteorology.

First, we use two normal predictors: the local predictor and global predictor. The local predictor uses temporal data sets to predict the rough result as a baseline, which only takes the usage record from itself into consideration and will not be influenced by outside stations. The global predictor calculates the influence from nearby stations, such as the direction of bike flows, relative distance, and other spatial correlation. The two predictors are combined at the end of this process, where different predictors play essential roles in very different cases. For example, usually, the bike records follow the result of a local predictor. When a traffic flow comes at the closing time, it increases the usage of nearby station significantly. Notice that the reason why we divide the predicting process into two parts is the distinction between temporal features and spatial features. They have no overlap with each other. Besides, the prediction based on a mass of temporal features is mainly trained by linear regression, in which the computation cost is relatively low. The spatial features are less but are complicated enough for learning model [19]. With the integration of these two predictors, all the data sets will be trained more accurately than putting all of them into a single learning model [56]. When the weather of strong storm or low temperature comes, the usage ratio of each station will drop tremendously for a while. This kind of drop is easy to be detected but hard to be trained as a regular model [55]. Therefore, these individual cases should be picked out and trained by another predictor. Then, we analyze the sudden drop instances based on the observation and develop a classification method that minimizes the variance while keeping most of them as ordinary cases [42], where a series of thresholds describes the classification method. This process of inflection detecting and predicting model is triggered when one of these features is beyond its threshold. As mentioned above, these unusual cases only influence the results for a few hours but change the line sharp significantly. We treat them as another data set of bike usage records and train them separately. When the inflection predictor detects the inflection, it provides the calculated deviation before the final aggregating. The independence of this process can emphasize the impact of extreme conditions, which improves the accuracy of the whole process. The process of predictor aggregating integrates the predictions of preliminary data and inflection data. This hybrid system predicts the result in future time intervals. For the sake of understanding, we divide the prediction of future usage into hours, from 1 to 6 hours and one day in the future, respectively. The final prediction stores in a maintenance monitoring server.

Maintenance Service: In real maintenance works, the schedule depending on real-time data is not efficient enough since there are relatively long delays before the replenishment fleet arriving at the target station. Due to the delay, most of the low battery bikes maybe out of the station, which makes the current data based schedule meaningless [9]. Instead, depending on the prediction of history data, the system follow the pattern of pedelec traffic flow and outputs the schedule. We propose an algorithm that calculates the optimized routes for replenishment trucks to exchange empty batteries. The maintenance monitor advises the administrative department with new updated predictions. The new records are updated in every time interval, then our system calculates the replenishment schedule dynamically. When the real-time usage record is very different from the prediction, usually the difference is more significant than a threshold, we have sufficient reasons that there may be some emergency situation or accident in a specific area. The emergency monitoring server notifies the administrative department of the city. Meanwhile, our system calculates the escape route if necessary.

5.2 Hybrid Prediction Model

In this section, we discuss how the local predictor, global predictor and inflection predictor work to create predictions results, and finally gather them together using prediction integrator.

5.2.1 Local Predictor

Local predictors are primarily based on the characteristics of station itself to predict future usage. They are quantitatively analyzed for environmental and land form disturbances. The local predictor trains the temporal data sets including: 1) the usage record in the past days at the station; 2) the local meteorology (such as temperature/precipitation/snowfall/wind speed, or certain types of weather like Thunder/Ice Pellets/Blowing Snow); 3) calendar related information (holiday/weekend and workday); and 4) weather forecast information.

These features have correlation with the bike usage by the observation of our raw resources. All these features are formulated into numerical matrices, and are used as the original numerical format of National Weather Forecast Office of U.S.A: the data sets indicating one particular state of a period, such as thunder, ice pellets and even holiday, are formatted as binary variables, e.g. $foggy = 1$ indicates this period is foggy; and other continuous data sets, e.g., temperature, precipitation, keep in standard matrices. The windspeed is the average wind speed in miles per hour (mph). The precipitation is the total precipitation in one day to the nearest hundredth of an inch. PSBL is the percentage of possible sunshine, which sensed by dividing the minutes of sunshine by the total possible minutes; Some data sets related to human-made activities rules, are described by different binary variables even they belong to the same category. For example, Monday is zero and Saturday is one if the system wants to predict the traffic trend on Saturday.

Based on the input temporal data sets, the prediction on future few hours is computed. Here, we pick up the data set in the recent few days as the input. Since time is passing, the input of training sets changes every hour. We predict the recent six hours. For the next two days, we predict the intervals of each six hours. The reason why we use interval instead of hour is that the predictions of recent six hours are strongly influenced by the current data when the prediction of 7 to 48 hours roughly follows the historical trends.

Next, we discuss the details of linear regression modeling here. At first, as introduced in Section 3, we convert all the features into numeric documents: boolean values are changed into (0, 1) to represent the extreme weather (storm/thunder/snow); All of the date information is changed into timestamps and 1 to 7 to present the weekday. Notice that the already predicted results will not be put into the modeling process again since it may cause the iterative error due to the previous predictions. Also, the numeric weather forecast data set may change periodically, which affects the bike usage prediction distinctly.

5.2.2 Global Predictor

Not only local needs affect the usage of bikes, but the traffic flows from nearby stations also impact on the bike demand [51] [49], and a global predictor could predict it. The motivation for using a global predictor is that there is some connectivities between specific stations. Usually, a small surge in usage also affects the surrounding stations in the future. We infer indirectly the growth of bike usage based on the surrounding stations.

According to our statistics, users keep a speed of 10 to 15 km/h by cycling and 86.4% of the duration time are below 20 minutes (in 2016/04). The service condition of nearby stations plays an essential role in the prediction of the current station, which should be taken into consideration. To understand the geographical connections between stations, we develop a Bicycling Sharing Radar Cross Model (BSRC) to describe the nearby bike usage conditions. Fig. 5.4a shows the BSRC which represents the bike usage of nearby stations for station 491 (coordinator [40.741 -73.983]) in 2017/03. We divide the radar plane into fan shapes, noted as $Q_1, Q_2, Q_3, Q_4, \cdots$. The number in each fan shape means the number of renting records in this station. In one fan shape, we divide it into three regions based on the distance ([0, 500], [501, 1000], [1001 ∞] meters). Each region contains the sum of bike usage in the same fan shape with the specific distance range. The three different kinds of color (green, yellow, red) means the different number of levels ([0,20], [21 40], [41 ∞]). From the figure, we can find that the bike renting ratio in region 1 of fan shape q_1 and q_8 is relatively high since there are public service facilities (hospitals, etc.) within 500 meters. However, region 3 in q_8 is small since most of this region is water area. The Fig. 5.4b shows the BSRC of returning record in station 491 at the same time. Through this BSRC, the nearby stations can be regarded as traffic flow sensors to support the prediction of the target station.

Following are reasons why we turn associative zone to a BSRC and integrate the spatial data sets into the definition of regions:

1. It is impossible that we put all stations' data into the learning system individually. The server can not afford the cost of computation when facing the ever-increasing usage demand [15]. At the same time, the time cost also prevents the real deployment of an online service. The BSRC decreases the number of inputs significantly, and the training model can save the time that learns the geographic association by itself. The only thing we need to adjust is the number of regions m.

2. We use the total usage amount instead of the usage record that flows to the target station because it cannot reflect the real traffic flows. For instance, bad weather may cause a lower usage ratio of bikes. On the opposite, the lack of available bikes in peak time may also cause the decrease of bike usage.

3. Several pairs of stations have no direct path due to the river and highland. Therefore, these stations may become redundant in the prediction process. However, these stations still can reflect the popularity of sharing bike system during a specific period.

4. The adjustable number of fan shapes provides the adaptability in different cities. Different sizes of angles represent the sensitivity of this model from directions, and different lengths of regions show the gradients that influenced by distance.

After aggregating the spatial data into regions, we form the mentioned BSRC into arrays with the time series. Besides, not only the recent usage record is formed, other notable features are also considered in the BSRC: 1) weather features in the specific regions, including wind speed, humanity, temperature, precipitation, snowfall, minutes of sunshine, other unusual weather conditions (Fog/Thunder/Ice pellets/Hail/Glaze or Rime/Blowing Dust or Sand/Smoke or Haze/Blowing Snow/Tornado). 2) public service facilities, including those used for educational (schools, libraries), recreational(hospitals, parks, theaters, stadiums), and cultural (museums, memorial halls) buildings.

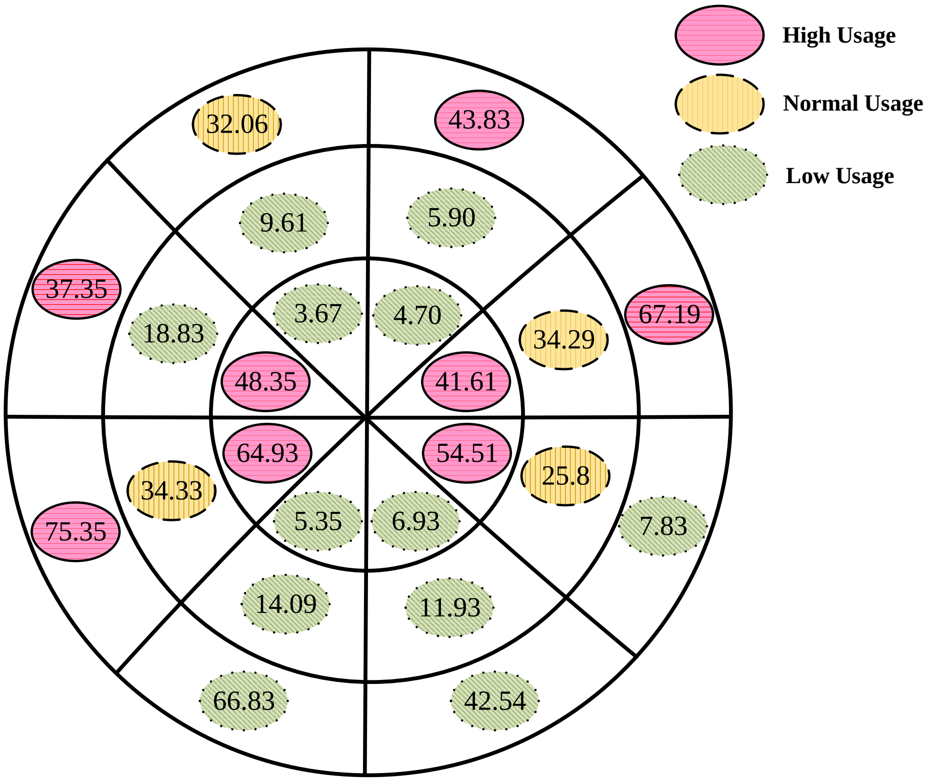
Fig.5.5 shows the structure of input data and ANN for prediction. For the sake of simplification, all the datasets are formatted into numbers. The usage number of bicycle start $Start_x$ and return End_x records are counted based on the updating. The $Weather_x$ indicates the average temperature among these stations, and each station counts other parameters that represent states in this fan. For example, Pub_x states the number of public service facilities, which presents the importance of residential activity center for citizens' daily activities. As the same, the statistical results from the same fan are input to the full-connected neutral network group by group, with the order of i divided fan shapes and j distance based regions. Similar to Section 4.1, we predict the future hours based on the interval of six hours, respectively. Notice that the format proposed above is just an example in one interval (six hours), here we input the data within recent one week.

Then, we feed the data set into the 3-layer artificial neural network, which the time cost and accuracy are acceptable after adjusting several parameters. The output is the increment of bicycle usage $\Delta Usage$ comparing with the 'current' number of bicycle usage. Then, we can predict the usage in future k hours $Usage_{t+k} = Usage_t + \Delta Usage$. This value is also estimated by a threshold to detect the sudden change of bicycle usage in the city, which discussed later.

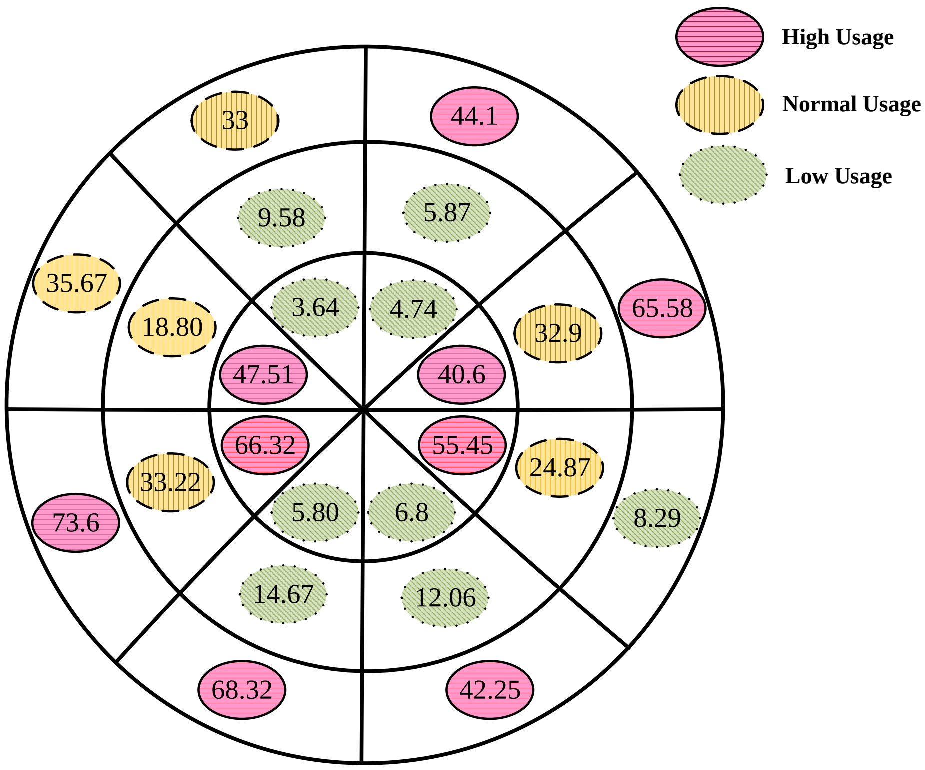
5.2.3 Inflection Predictor

An inflection predictor is proposed here to handle these sudden changes. Generally, The usage record of a station may increase or decrease sharply during a few time intervals due to the sudden changes of terrible rainfalls. Although we can filter these unusual situations to make a genetic one, and applies to most situations, they genuinely show the apparent influence to people's daily life, and moreover, the reaction in emergency situations.

For instance, the heavy rain and strong breeze can stop most citizens' bike travel. As a result, the usage records in some stations keep 0 through one day. These kinds of significant curve deformation we call them inflection situations, and then we extract these data sets from the normal ones by the following steps:



(a) A shared bike radar cross Model of station 491 (start)



(b) A shared bike radar cross Model of station 491 (end)

Fig. 5.4 An example of BSRC. The red circle is high usage (> 40), yellow is normal usage ($20 \rightarrow 40$), and green is low usage (< 20). The distance of regions is classified as 500, 1000 and infinite in meter.

New Updated Dataset

1	$Start_1$	End_1	Wea_1	**......**	Pub_1
2	$Start_2$	End_2	Wea_2	......	Pub_2
......	......	......	......	......	......
i+1	$Start_{i+1}$	End_{i+1}	Wea_{i+1}	......	Pub_{i+1}
......	......	......	......	......	......
ij-1	$Start_{ij-1}$	End_{ij-1}	Wea_{ij-1}	......	Pub_{ij-1}
ij	$Start_{ij}$	End_{ij}	Wea_{ij}	......	Pub_{ij}

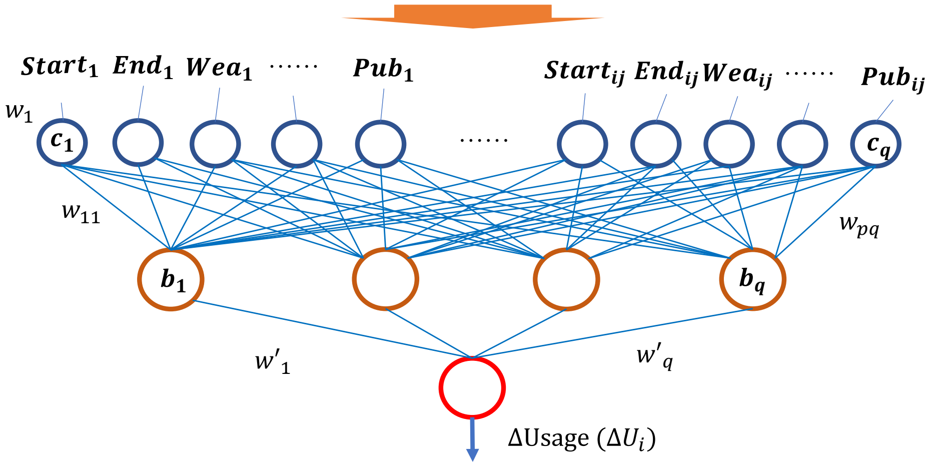


Fig. 5.5 Structure of input data set and a 3-layer artificial neural network for prediction. The table is an exsample of format for each input samples. The network structure shows the connection between input of train set, hidden neurons and output of future changes

1. Make the rough selection to separate the inflection dataset S' from the whole data sets S : At first, we set a threshold based on common sense to point out the unusual usage record. In a case of station A, if the number of records in one hour is 200% bigger or 10% than the average (603 and 30, respectively), we consider these records as the inflection dataset S' which may infect the usage curve deformation massively.

2. Find the notable sudden change ' interval': A sudden change interval for a feature (e.g., temperature) is a continuous numerical interval that the distribution of set S' in this feature is higher than the distribution of set $S - S'$. For example, if the interval of temperature is more than 35 degree Celsius until the infinite, it means the usage distribution of set S' is higher in this interval and lower than $S - S'$ in other degrees. Similar with Fig. 5.6a, We plot the probability distribution of set S' and the complementary set $S - S'$ for each feature. Then we can observe the intervals from features that the probability distribution in dataset S' is greater than S. These intervals are considered as the distinctive sudden change intervals for describing inflection data

(or the distinctive category for discrete feature). When the feature of one new record falls into the distinctive interval, we can classify this record to dataset S' and active the inflection predictor.

3. Assign the distinctive intervals as thresholds. The range of multiple distinctive intervals of a feature may be suitable for a given data set S'. However, we still need to find the precise connections if one feature holds multiple intervals. To find the precise set of distinctive intervals, we define an environmental fitness Q to form the notable interval selection problem into a coverage ratio optimization problem, where we have

$$Q = \frac{|D(y') - D(y)|}{|D(x') - D(x)|} \cdot \left(\frac{|x|}{|S'|} - \frac{|y|}{|S - S'|}\right) \tag{5.1}$$

where the absolute value signs $|S'|$ indicates the number of records in S'. x and y are the records belong to S and $S - S'$. $D(x') - D(x)$ indicates the difference of data square error before and after adding a new given set of distinctive intervals. Through the fitness Q, we can search for the notable interval set which covers most of the inflection records and affects them. This problem can be solved by a genetic algorithm or particle swarm optimization, which depends on the data scale.

4. Train an inflection predictor with the highest fitness interval set. After finding the highest fitness interval set, we update the inflection dataset S', which could satisfy all the thresholds (or intervals) and discrete features (e.g., thunder). Then we put the dataset S' back to local predictor and global predictor train a new hybrid predictive model independently, the process is similar to the mentioned predictor.

For training such a model, the input is the updated dataset S' and other environment information. Notice that not all the usage records are included in the original defined set since the original definition is developed to discover the inner connections of the strange usage phenomenon and general standards for handling inflection and sudden changes. Therefore, the collection of input data set is the usage records satisfied by the mentioned constraints, and we use another paralleled process of the former predictor (local and global predictor) to calculate the results, where we could carefully keep the consistency with the previous process. To distinguish the difference, we claim as the inflection predictor, and the output is the increment of bicycle usage $\Delta Usage$, comparing with the defined 'current' data set. When the time goes, and newly updated records come to the prediction system, it estimates whether they fall into the threshold and trigger on the inflection predictor if necessary. The system separates the records in two sets, $S - S'$ and S', and fed them into regular predictive precess or inflection predictor, respectively. The purpose of the additional means is that the predictive model is a response mechanism that could handle the sudden changes smartly, including but not limited to the unusual situations like disasters and large-scale activities. By inputting other environmental information such as weathers, they predict the results from each predictor. Finally, all the predictive results of $\Delta Usage$ are combined in prediction integrator to general a more precise result.

Based on the equation proposed in page 6, we find the three features of temperature, perception and wind speed have an apparent difference of distribution using different sets. However, when processing these unusual situations, based on our results in Section 5.2, the accuracy is not as high as a usual predictor, where we consider our inflection detection method could only cover part of the sudden change phenomenon. Still, it could help improve the accuracy in final result comparing with usual predictor only processing.

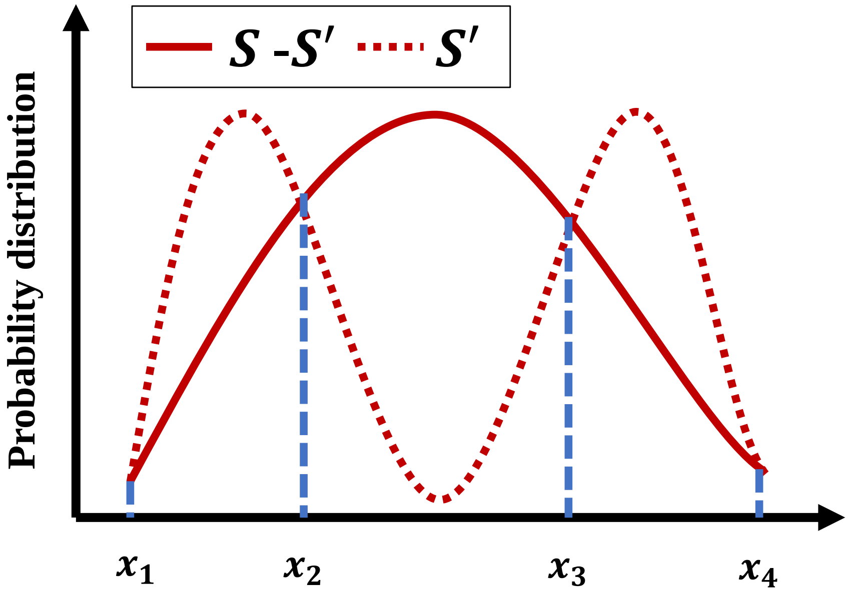
5.2.4 Prediction Integrator

The prediction integrator combines the results of three predictors. By observation, we find that usually, the local predictor plays an essential role if it is serenity. However, the global predictor affects the prediction significantly when there are large-scale activities nearby. As mentioned before, lousy weather affects humans' intention of bike traveling. Therefore, we use a decision-making tree as the final step of our prediction process. Fig. 5.6b shows an example of a decision-making tree, and both continuous and discrete features are considered in this process. To process the continuous features, we use the influence of variance to detect the partition thresholds. The feature which has the most significant decrease of variance will be selected as the first ramification point of the proposed discussion-making tree. Then, the process estimates the other features in the next two separated branches and continues to create branches. Finally, it generates a level limited decision-making tree, where the characteristics of the model determine the elastic number of levels. Here, we only find the results from the three predictors have the relatively high influence on the different leaves, and the features of weather are next.

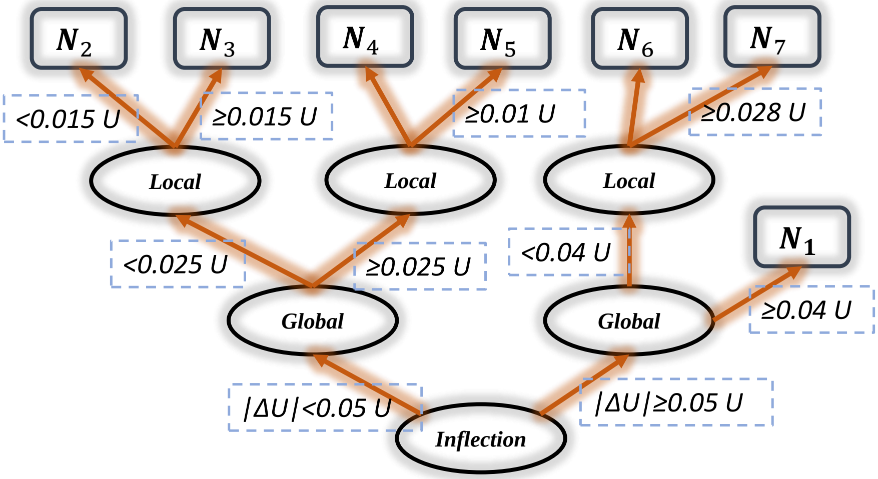
For a specific leaf i in the decision-making tree, we use a linear regression of bike renting usage N_i and returning usage N_i' to combine these prediction results and features, where we have

$$N_i = a_{i,1} \cdot Local + a_{i,2} \cdot Global + a_{i,3} \cdot Inflection \tag{5.2}$$

where a_i indicates the different coefficient impacted by predictors and weather features. We use a simple linear regression to train these coefficients for each leaf N_i. For training these



(a) An example of finding the sudden change interval $x_1 \to x_2$ and $x_3 \to x_4$ (the dash line shows the probability distribution of inflection data set and full line show the usual dataset)



(b) A shared bike radar cross Model of station 491 (end)

Fig. 5.6 An example of sudden change interval and decision making tree

coefficients, the input is considered as the prediction results from each predictor, and output is considered as the real increment comparing with current usage. Based on our observation, the prediction result from local, global act as the similar behaviors and inflection predictors perform relatively higher coefficient when the weather seems unusual (e.g., the temperature is high). Other features (Wind-speed) make the prediction a fine-tuning.

Above all, the three predictors have no overlap and can predict the bike usage independently. Next is the pseudo code of the overall prediction process. When new usage records updated by the online service, the system predict the usage in future time intervals immediately using the already trained prediction model. At first, the system estimates whether it is necessary to trigger on the inflection predictor based on the future environment information (e.g., weather forecast), then put the updated data set U into the predictors. Finally, through the trained decision making tree and different coefficients in it, the integrator combines these predictive results and emphasizes different increments to meeting with the complicated and fickle traffic environment as the time goes. The Algorithm is :

Require: U: updated new bike usage record from the service; t_{max}: the number of targeted predictive results based on different time interval; C: thresholds to trigger on the inflection prediction process; $Info_t$: other environmental information in future time interval t;

Ensure: Predictive result of usage increment $\Delta Usage$;

Array $\Delta Usage = \{0\}$;
for $t = 1 : t_{max}$ **do**
 if $Info_t$ satisfies one of the conditions in C **then**
  Input U to $Inflection\ Predictor$ and get $\Delta U_{t,3}$;
 else
  $\Delta U_{t,3} = 0$;
 end if
 Input U to $Local\ Predictor$ then get $\Delta U_{t,1}$;
 Input U to $Global\ Predictor$ then get $\Delta U_{t,2}$;
 Active Prediction Integrator to combine the results $\Delta U_t = Integrator(\Delta U_{t,3} + \Delta U_{t,2} + \Delta U_{t,1})$;
end for
Update the prediction array ΔU using ΔU_t;

5.3 Pedelec Maintenance Optimization Algorithm

The usage of renting and returning, bike storage of each station are predicted through above process, and we use these predictions to design a feasible optimization algorithm for every replenishment truck, to exchange the batteries at each station.

We divide our solution into two steps: (1).Construct a less incompatible maintenance map to offer suitable station-to-station links; (2).Reduce the dynamic replenishment problem into a

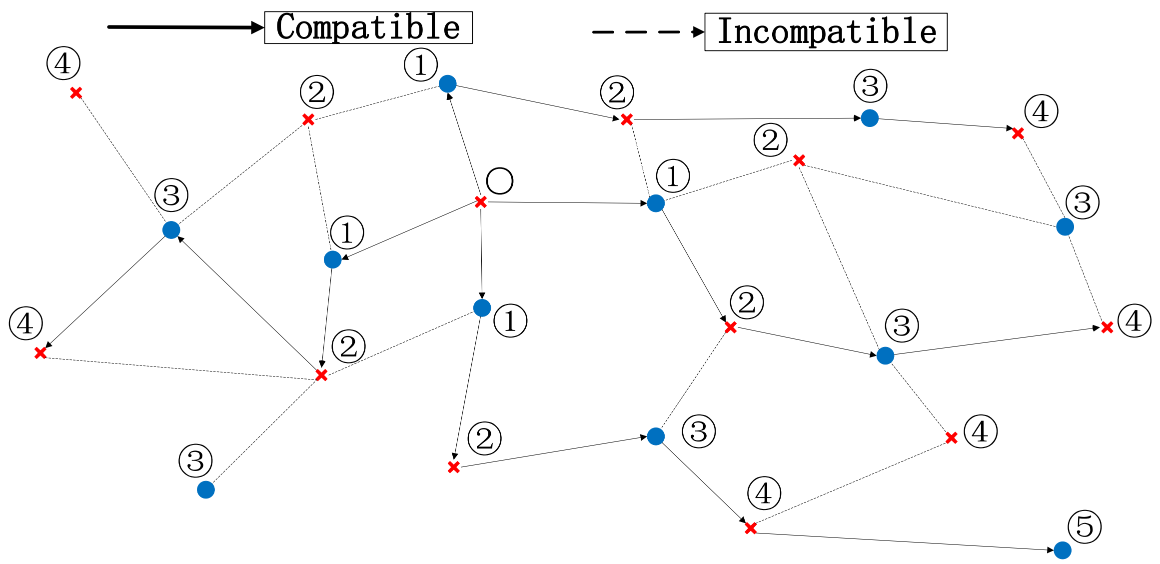


Fig. 5.7 An example of less incompatible maintenance map (solid and dashed lines are compatible and incompatible links between stations. Numbers from 1 to 4 present an feasible route for one truck.)

classical TSP. (3).Develop a heuristic algorithm called intelligence sharing bike maintenance optimization algorithm (IMOA) to improve the performance.

5.3.1 Less Incompatible Maintenance Map

As discussed earlier, it will cause the overlapped problem if several trucks go around in the same area. The bikes may happen to follow the same route with the replenishment truck and served twice[39] [52].

For such a problem, we set an indicator $r_{i,j}$ to indicate the proportion of returning at station i from the renting station j. Then we set a threshold $r \leq r_{i,j}$ to indicate a less incompatible link from station i to j. As shown in the Fig.Fig, imaginary lines are the incompatible links that are closed to each other and may cause overlapped problems. We use the average distance of the collected distances of each most extended incompatible station, noted as l_1. Considering the time cost on the road, and there are more than 400 stations in the city, our replenishment truck complete more than k maintenance mission within an hour. We set a maximum range for each mission. Therefore, the trucks can guarantee k missions within one hour, noted as l_2. Therefore, we can construct a less incompatible maintenance map to indicate the candidate links (distance $\in [l_1, l_2]$) from stations to stations that the trucks can follow. The stations that nearby enough (less than l_1) are considered as incompatible links and the stations that far away are considered out of truck's moving range.

For instance, Fig. 5.7 shows an example of less incompatible maintenance map, where the solid line is the feasible links between stations and dashed lines mean the indicator is too high for constant maintenance. The numbers indicate the feasible steps of each route and linked by the stable route. Through it, the system can narrow down the optional range of next stations and reduce the computation time.

5.3.2 Reduced Dynamic Replenishment Optimization Problem

After making a less incompatible maintenance map, we design routes for trucks that cover lower battery remaining stations as many as possible. We assume there are w replenishment trucks from the warehouse and $s \in G$ stations are available for bike maintenance. We consider this warehouse as an origin point, which considered as an additional 'station' on the map. Therefore, this problem is reduced into a traveling salesman problem (TSP): Given a map list D of vertices(stations) and arcs(links) between each pair of cities, find the optimal route that satisfies maximized demands D among all the cities.

Different from the classical static TSP problem, the replenishment demand of each station is continuously changing [63]. Although the calculated schedule may be the optimal solution in current period [20], however, as the time goes by, the solution cannot fit with the demand change in each station. One potential solution is the genetic algorithm, which is a bionic optimization algorithm and has the capability of random large-scale search [40]. The calculation speed is adjustable and comfortable to combine with other algorithms, and is suitable for static TSP problems [17]. However, the feedback in the system is not enough in our dynamic problem. Furthermore, when the demand of each station floats, it often creates a lot of inaction redundant Diego generation and may enter the local value, which makes the efficiency of the system very low.

With the observation above, combining the advantages of ACO (Ant Colony Optimization) and GA [24][7], we propose an intelligent pedelec maintenance optimization algorithm, which is based on the pheromones on the links left by the former trucks [45], to improve the efficiency of the whole system. This algorithm can make trucks no longer confined to the pheromones when selecting the path, and make the use of existing information and search new path synchronously. It correctly increases the randomness and diversity of solutions and dramatically enhances the robustness of the basic heuristic algorithm.

To fit this specific problem, we describe the TSP problem again: There is a total number of n stations in the city, noted as $(1,2,\cdots,n \in V)$. One truck departs from station 1 to start its tasks, where the probability travels to city i is noted as p_{1i}, and the probability travels to another city p_{1j} is independent. No matter how many time it takes at the station i, the time serving in other cities will be delayed by t_{1i}. For instant, one truck starts at station 1, then $p_{11} = 0$ and $t = 0$. It costs time t_{ij} for the truck travel from station i to station j, where $t_{ij} = vd_{ij}$ and v is average speed of replenishment trucks. If the task at station i is completed, the truck travel to another station k with the probability p_{ik}. The optimization goal is that we design a schedule which contains the stations with the highest expected number of lower battery bikes.

Require: $G = (V,A)$: created incompatible maintenance map contains vertex and arc; w: number of replenishment fleet; U: prediction results of bike usage; q: set qth best replenishment schedule;

Ensure: Schedule P;

Set incompatible indicator array $\sigma = IncIn(U)$;

Set travel probability array $p = TravelArc(z_0, \sigma)$;

Pheromone array $z = z_0$;

for each $g = 1; g \leq Generation; g++;$ **do**

    $Path = \{0\}$;

    Temporary Station Set $V' = V$;

    for each $i = 1; i \leq w; i++$ **do**

        $Path_i = RouteSelection(V', p)$;

        Delete stations in $Path_i$ from V' ;

    end for

    Sort $Path$ and Select best qth schedules, $Path'$;

    Pheromone array $z = UpdatePhe(z, Path')$;

    Travel probability array $p = TravelArc(z, \sigma)$;

end for

$P = Path$;

$t = 0$;

while $t \leq t_{max}$ **do**

    Input Updated $G = (V,A), w, U$;

    $P_t = AlgorithmII(G, w, U)$;

    $t = t + \Delta t$;

end while

    return P;

5.3.3 Intelligent Maintenance Optimization Algorithm

Next, we take the demand of each station into consideration. Trucks tend to select a path with higher demand, lower distance and higher pheromone provided by former trucks. Our prediction process predicts the demand for replenishment task, and pheromone is the pattern that describes the relativity of one link in a traditional optimized solution.

According to the above concerns, we define the probability that the truck travels to another station as,

$$p_{ij} = \begin{cases} \dfrac{z_{ij}\sigma_{ij}^{\beta}}{\sum z_{ik}\sigma_{ik}^{\beta}}, & \textbf{where} \quad i,j,k \in V' \\ 0, & \textbf{where} \quad i = j \end{cases} \tag{5.3}$$

where z_{ij} indicates the pheromone of the link from station i to station j and set a parameter z_0 as an initial value. β is the adjustable parameter and $V' = \{1,2,\cdots,n\} - V_0$ is the candidate stations, where V_0 is the set of stations served recently. σ_{ij}^{β} indicates the importance of stations' maintenance order, which try to find the station with empty battery bikes as many as possible. The importance σ_{ij}^{β} we define here is,

$$\sigma_{ij}^{\beta} = \frac{U \cdot S_r}{S_b \cdot t_{ij}} \cdot \frac{1}{c \cdot CS + d \cdot MS} \tag{5.4}$$

where U, S_r, S_b is the predicted bike storage, renting ratio and returning ratio. CS is the number of nearby served stations and MS is the number of nearby marked stations that are planned to be served. c and d is the adjustable parameter. t_{ij} is the time cost if travels to station j. This σ_{ij}^{β} considers the possibility of dry battery bike storage in current stations, as well as the spreading of bikes before and after maintenance.

The algorithm allows the trucks to find the optimal path for updating the pheromones, and tehen this parameter is updated globally, noted as

$$z_{ij} \leftarrow (1-\phi) \cdot z_{ij} + \phi \cdot \triangle z_{ij} \tag{5.5}$$

where $\phi \in [0,1]$ and $\triangle z_{ij} = D_{ij}$ which indicates the total number of served bikes at station j using the journey i to j.

However, before the system converges to global optimal solution, the current solution may fall into the local minimum, which leads to a stop of iterations. In order to avoid this, we update the rule of pheromones. At each iteration, only trucks with the best solutions can update z_{ij}. We select q trucks with the best q solution, and the definition of pheromones becomes,

$$z_{ij} \leftarrow (1-\phi) \cdot z_{ij} + \sum_{q=1}^{m} \phi \cdot \triangle z_{ij,q} \tag{5.6}$$

where the new $\triangle z_{ij,q}$ we define here is,

$$\triangle z_{ij,q} = \begin{cases} \dfrac{D_{ij}}{q}, & \textbf{where} \quad i,j \in V' \\ 0, & \textbf{where} \quad i = j \end{cases} \tag{5.7}$$

where q indicates the qth optimal solution in current system.

When the new updated of prediction results come, firstly, the system modifies the incompatible map and reset the incompatible indicator array σ and travel probability array p. Then, set the initiate pheromone array as z_0 and enter into the iterations process. The main steps of iteration in the algorithm are described as the following: (1) We set w trucks and input the current prediction result. These trucks travel to another station following the less incompatible map and use Eq.5.3 to select the next station. (2) When all the trucks finished their journeys, the system sorts the qth best solution and uses Eq.5.6 to update the pheromones Δz on the arcs. (3) Reset all the trucks into the initial state and update the travel probability p between stations using new pheromones Δz. (4) Repeat the above steps until the number of iterations runs out. Finally, the system outputs the optimized route as P.

The above steps are the process in our proposed IMOA algorithm, where Algorithm 2 is the pseudo code. At first, we input the prediction results and construct a less incompatible maintenance map. Then, we use Algorithm 2 to train a model and calculate the detailed schedule for each truck. Finally, our IMOA outputs the routes with specific stations and time stamp for each time interval, using the dynamic variant routing algorithm (pseudocode in Algorithm 3).

5.4 Simulation

This section verifies the performance of our proposal using simulations. In each simulation, we apply the proposed pedelec maintenance system in a real-world city map. The settings of pedelecs and city big data sets are extracted from the open data of governments, then the data is input to the proposed system.

5.4.1 Settings

We use the open data of bike sharing system 'Citibike' in NYC for our simulations. 'Citibike' is a widely used data set that has already been visualized. Besides, the geographic information and weather information are also well discussed by other studies [4]:

'Pedelec' and Battery: As mentioned above, Pedelec is a kind of electric bike that is very useful when riding in hills or strong winds. Since there is a revaluation about the battery in these e-bikes, we consider three different types of power replenishment system in the simulations, lead-acid battery (or other conventional battery about 15 Ah), lithium-ion battery (e.g., 24 Ah) and lithium iron phosphate battery (e.g., 48 Ah) [38]. For an adult, he can keep the speed of 6 km/h without pedaling or keep the speed of 10 to 13 km/h within the range of 100 kilometers under the support of electronic controller [44]. Therefore, we assume one lithium ion battery can keep working for 6 to 8 hours. When a Pedelec needs maintenance, the replenishment truck can install a new battery on it instead of the empty battery if stopping at the station [1]. Since they all use rechargeable batteries, we assume one replenishment truck can replace all the batteries at the station.

'Citibike': It is a set of sharing bike records starting with 2013/12 and ending with 2017/04 in the website of Citibike, and we have shown the statistical result of total usage in Section 3.1. One essential record contains trip duration, start time & stop time, start station & end station, station locations, start time & end time, coordinator, etc. As counted in the introduction, there are more than 600 stations. We choose the data for this year (2017/03-2017/05) as our testing data, and the others are considered as training data.

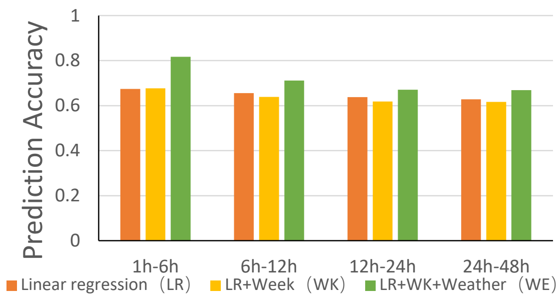
Weather Report and Forecast: We use the data set of National Weather Service Forecast Office for our study. One essential record in a specific area (e.g., central park) contains the date, max & min & average temperature, precipitation, snowfall, direction & speed & peak of wind, particular weather (fog, thunder, tornado), etc. The most exciting thing is that compared with other traffic flows (such as cars or metro) [23], the activities of bike traveling are more sensitive to the weather condition. In another word, lousy weather inflects our 'human sensors' significantly.

Geography Information: Google API is applied in our study to gather useful geography information that helps the system understand the geographical correlations between stations.

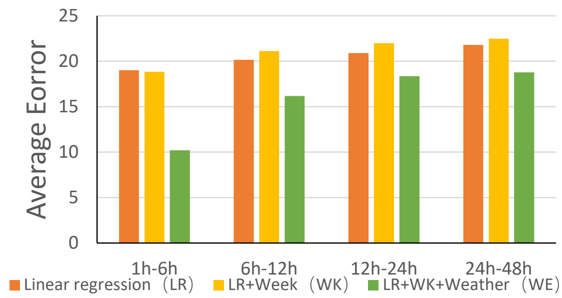
5.4.2 Accuracy of Predictor

First, we analyze the accuracy of each predictor and discuss how a citywide big dataset performs for the pedelec maintenance system. Fig. 5.8a and Fig. 5.8b shows the average probability and error of bike usage at each station. We input 15 days' data and predict the bike usage in recent two days. The linear regression only method reaches the accuracy of 67.39% within recent 6 hours, where the accuracy is defined as,

$$\overline{A} = 1 - \sum_{i=1}^{n} (\overline{U} - U_i) / \sum_{i} U_i \tag{5.8}$$



(a) Average accuracy in each station

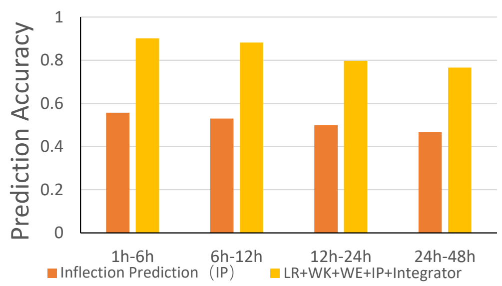


(b) Averaged error in each station

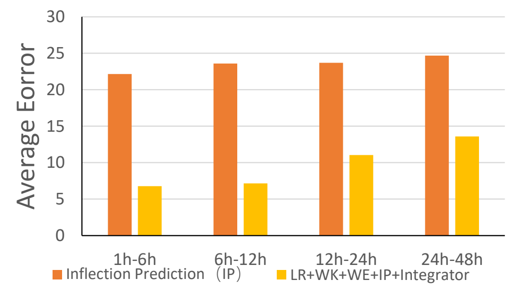
Fig. 5.8 Local predictor using citywide big data

where U indicates the number of bike usage record. However, the additional week data (noted as 1 to 7) shows no help for the prediction, which means no strong connection with bike usage. The same with the observation in Section 3, the weather information shows important influence on the prediction, and it reaches 81.68% within new 6 hours. The average error is defined as,

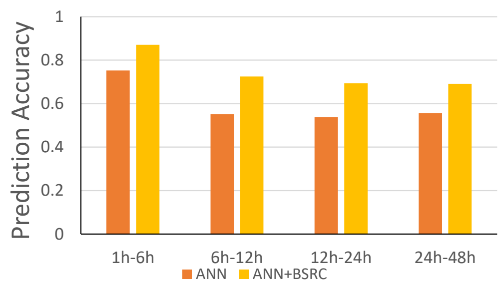
$$\overline{E} = \sum_{i=1}^{n} (\overline{U} - U_i)/n \tag{5.9}$$



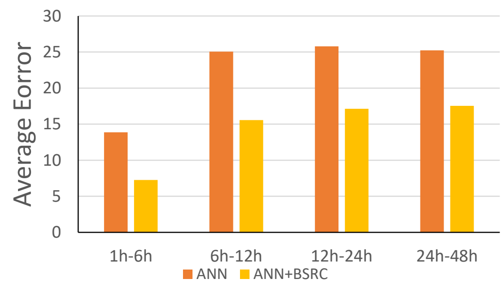
(a) Global predictor using citywide big data (average accuracy in each station)



(b) Global predictor using citywide big data (averaged error in each station)



(c) Inflection predictor and integrator using citywide big data (average accuracy in each station)



(d) Inflection predictor and integrator using citywide big data (averaged error in each station)

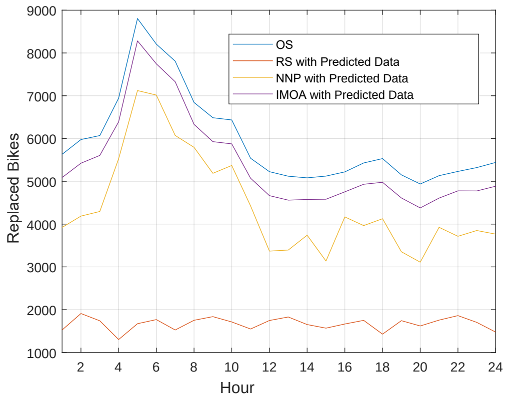
Fig. 5.9 Comparison of different prediction model

It shows the prediction result is below 10.200 for all the stations, while linear regression only method and additional week data can help reach 19.011% and 18.846%, respectively. Fig. 5.9a and Fig. 5.9b show the prediction result of global predictor. We input 15 days' data and use an ANN to learn the result. The historical usage data reaches an accuracy of 75.26%, which is better than linear regression only method. Then, we use additional global information from other stations and form them in the BSRC, and the accuracy of the results reaches 87.12%. The average error of $ANN+BSRC$ is 7.275 within new 6 hours, which is the half of ANN only prediction (13.86). Besides, both the average error and accuracy look wired between 24 hours and 48 hours. We set the number of neurons in a hidden layer around $s = log(2N)$, where N is the number of neurons in the input layer. To keep it real, we use one parameter setting that the accuracy in 6-12h is the highest (6-12h is more meaningful for real deployment). Based on our observation, there is no big difference for the future 6-12, 24 and 48 hours as the predictive result for ANN. Besides, it also a powerful piece of evidence that our system design could avoid accumulated errors from previous predictions. The system could predict more extended future hours parallelly with the near future, without interference with each other, which is also one of the advantages. Fig. 5.9c and Fig. 5.9d show the accuracy of our inflection predictor and integrator. Since the inflection predictor processes the sudden change data sets, including about 10% of the total records, The upper and lower limits of the result are very high, therefore, the accuracy only can reach 55.68% about new 6 hours and the average error is 22.15. However, the prediction integrator successfully combines the results of the local predictor, global predictor, and inflection predictor, and the accuracy is 90.11%, 88.2%, 79.69% and 76.58% in recent 48 hours. The average error, which is a good result that, the prediction with averaged 6.78 error is approached, and keeps it below 15 within recent two days.

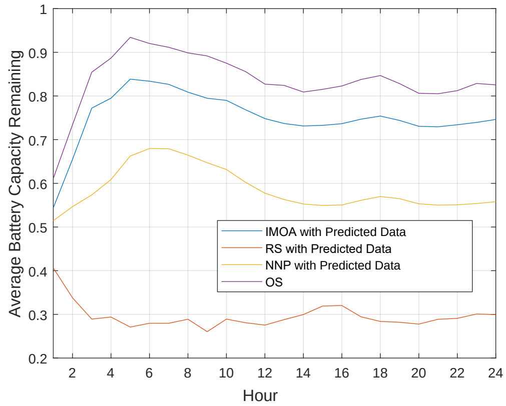
5.4.3 Results Comparisons of IMOA

In the experiment, we apply four maintenance strategies to evaluate our system. One strategy is the optimal solution that assuming all the future usage records have already been foreseen, and replenishment trucks ignore distance to travel immediately (noted as Optimal Solution, OS). IMOA method uses our proposed algorithm and the predicted results. Similar with [**?**], the third algorithm uses the standard nearest neighbor procedure (NNP) with highest bike storage. The fourth one completes replenishment tasks at stations without strategy (randomly selection, RS).

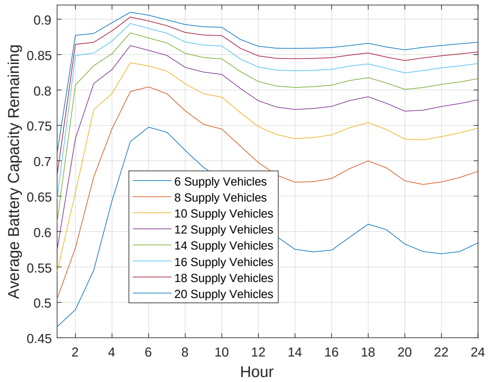
Fig. 5.10a, it shows the number of charged bikes during a day in 2017/04. For simplicity, we assume the replenishment trucks exchange all the batteries of bikes at the stations and then ignored the charge time. All the Pedelecs start with a random state of charge which



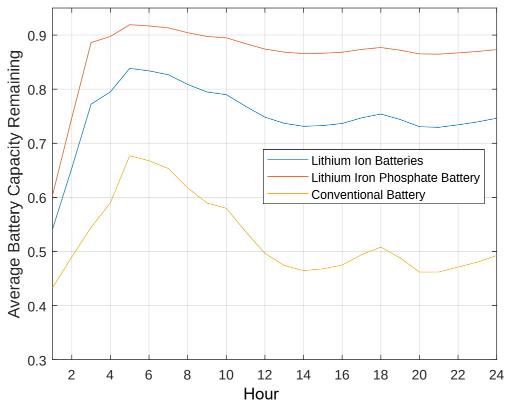
(a) Battery replaced bikes changed by the hour in a day (10 replenishment trucks)



(b) Battery remaining changed by the hour in a day (10 replenishment trucks)



(c) Battery remaining changed by number of replenishment trucks



(d) Battery remaining changed by different batteries

Fig. 5.10 Performance of four strategies by different settings

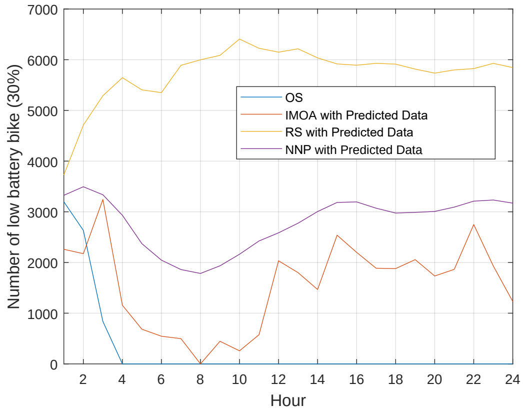
follows the uniform distribution. We assume that replenishment trucks are working during the day. The x-axis is the time series and we account every hour. The y-axis is the number of charged bikes, and 9765 bikes in 612 stations are monitored in that day. Except for the lowest one, we find the other three maintenance strategies can find the optimized route using their strategies. With the scatting of our replenishment trucks, the peak comes at 6 a.m. Another reason is the increment of traffic flows, while our IMOA algorithm based strategy can follow the traffic flows and stop at the stations with more bike storage. The next peak comes at 5 p.m., which is the closing time for citizens. From the figure, we can observe that our proposed algorithm outperforms the other two since we take the emission traffic trend of charged bikes into consideration.

Fig. 5.10b shows the average remaining battery using the same setting. The y-axis is the change of the average remaining battery among all bikes. Similar to the former figure, we can observe that our IMOA algorithm can detect the traffic flows smartly. However, the remaining battery can reflect the active maintenance. Therefore, the slope about remaining battery is more smooth than the number of charged bikes. Finally, the remaining battery stays at a moderate level as well as the others.

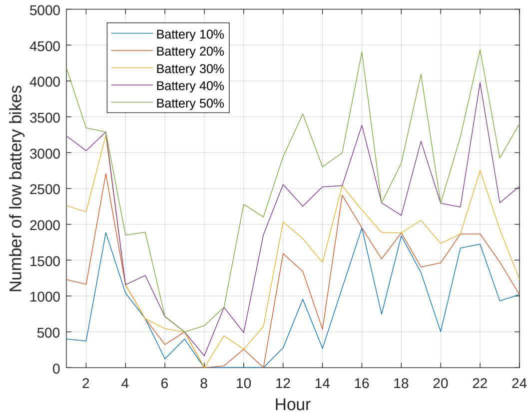
Fig. 5.10c and Fig. 5.10d are the average remaining battery using different settings. The former one is the average remaining battery changed by different replenishment trucks. The number of replenishment trucks is related to the maintenance cost. More replenishment trucks keep the average remaining battery of all bikes at a higher level. From the figure, we can say, our algorithm could significantly improve the average battery capacity remaining when the number of replenishment trucks is rare. Then, the growth trend decreases with the increase in vehicles, due to over more completable links on the city map. The latter one is about the different batteries. We assume all the bikes use only one type of batteries. The lead-acid battery can support average 3 hours while the up to date lithium iron phosphate battery keeps working for about 12 hours. Therefore, the remaining battery of the lead-acid battery cannot keep a higher standard since it runs down very quickly. The other two can finally keep average 76.3% and 88.7% batteries if the replenishment trucks keep working. The figure also shows the proposed optimization algorithm is indispensable for the old type of batteries since the demand for the battery change is more prominent. It also shows the advantage of new batteries, which has the strong robustness and lower maintenance demand in limited replenishment situations.

5.4.4 IMOA using Different Bikes

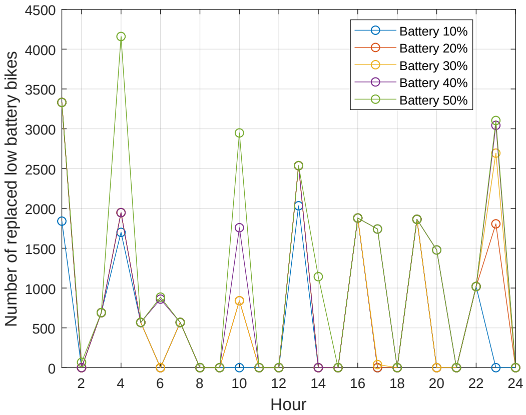
Next, we discuss the impact of low battery bike, which refers to the quality of user experience using these electric powered bikes. The primary purpose of the proposed algorithm is to keep the average remaining battery of all bikes at a high level, to ensure the normal operation of the share-bike system, and eliminate low battery bikes through our maintenance system since the power outage in use may lead to shorter travel distance and more human resources consumption. Therefore, the demand from each station that replacing low battery bike is a necessary evaluation standard in our simulations. Fig. 5.11a shows the number of low battery bikes using the same setting and algorithms. If the replenishment fleet uses the random selection method, averaged more than 6000 bikes keep the remaining battery lower than 30% and the total number of bikes in our system is 9765, which means 62.7% bikes need battery replacement within two hours. The nearby selection algorithm keeps the remaining battery at a lower standard and reaches the minimum due to the high traffic flow in the morning. Our proposed IMOA algorithm performs better than the former one due to the better maintenance schedule, and reach 0 minimum at the start of 8 o'clock. If we carefully analyze the curve of IMOA, as shown in Fig. 5.11b, we can observe that the different level of remaining battery



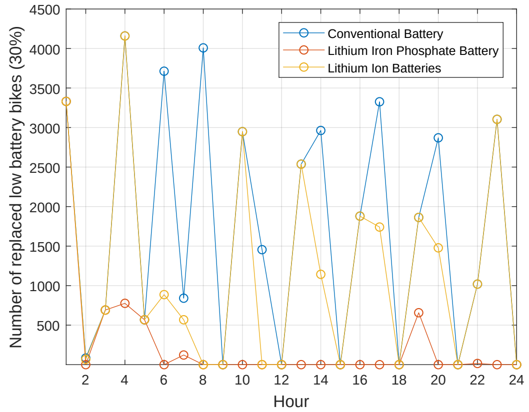
(a) Number of low battery bikes changed by the hour (10 replenishment trucks)



(b) Number of low battery bikes by the hour (10 replenishment trucks)



(c) Replaced low battery bikes changed by hour



(d) Replaced low battery bikes (different battery) changed by hour

Fig. 5.11 Performance of battery replacing demand

follows the same trend. Notice that the number of lower battery bikes (10%, 20%, 30%) reaches the same point at the 8, 18, 24 o'clock since the replenishment fleet patrols at the 'hot spot' stations in order to replace more batteries and some bikes in edge stations are ignored.

Fig. 5.11c shows the number of replaced low battery bikes for one day, and we distinguish the different level of the remaining battery to observe the result. As mentioned above, there are always demand the replacement of low battery bikes, and this figure shows how many demands have been satisfied through our IMOA. The higher number means more low battery bikes are satisfied, which creates the opposite slope with the former figure in most cases. The number of replaced bikes is also related to the number of trucks, and more trucks remain the battery ratio at the relatively lower standard. Fig. 5.11d shows the influence of different battery. Using the same setting as former one (10 trucks and same map information), the lead-acid battery bikes reach the 30% line more often than lithium ion and lithium iron phosphate embedded bikes. Through the figure we can say, in most of the case, the new lithium iron phosphate embedded bikes do not need to worry about the power shortage problem with a medium size of replenishment fleet. It also proves that only a smaller replenishment fleet is needed to reach the same performance as the normal battery, which greatly reduces the replenishment cost. In a word, our proposed algorithm performs better than the other classical solutions, and the average remaining battery keeps at a relatively higher standard. Besides, the demand for low battery replacing is also satisfied well, which make the whole prediction and scheduling system feasible.

5.5 Conclusion

This paper has studied the maintenance of pedelec system based on big data analytics, which meets the growing maintenance demand of new types. We have first studied the real-world urban data sets from all collect-able dimensions to explore the relations of different ingredients on the usage of sharing bikes. After that, a new hybrid predictive model to digitize the useful information in various fields and predict the future trend of bike usage has been developed. Finally, using the prediction outputs, a heuristic algorithm called intelligent share bike maintenance optimization algorithm (IMOA) has been proposed to solve the maintenance problem for replenishment trucks. Through simulations, we have verified our proposal from all angles and validate that the accuracy of our prediction model is high enough and IMOA solves the maintenance problem efficiently.

In the future, we intend to extend our work in two dimensions. First, the proposed system is feasible for NYC. However, the different landform and weather condition could significantly influence the bike usage and changes the inner connections between various datasets. We plan to develop a universal model to apply in different cites. Second, the efficiency and performance bound of the proposed algorithm have not been thoroughly investigated. We intend to use more artificial intelligence technologies to improve the performance of the whole system.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we investigate the exssting problem in urban ITS and achieve the transmission performance, user experience, system sustainability in urban intelligent transportation system.

6.1.1 Next-generation vehicle networking technology using Thz

To improve the performance issues, we study the impact of terahertz band towards vehicle networks. THz band communication relieves the spectrum scarcity and capacity restriction of the existing communication systems. In short range, the THz link can be considered as a transmission window with almost 1 THz, which supports the growing real-time data transmission. Moreover, we develop an autonomous terahertz relay algorithm called ATLR which gives the advice of ideal relaying position to bypass obstacles and avoids stronger NLOS fading. Finally, we apply this algorithm in autonomous vehicle network. It helps create more flexible communication environment and provide more traffic information to nearby autonomous vehicles. Actually, not limited to autonomous vehicles on the road, terahertz band can be widely applied to all mobile nano-cell networks, such as delivery drones or UAVs (Unmanned Aerial Vehicle). Different kind of 'vehicles' has very different ability to adapt communication environments. These 'vehicles' support the exploration of missing details, which helps human beings and human made machines understand the known world more smartly.

6.1.2 User-oriented taxi sharing service using social network

To improve the experience issues, we study the problem of taxi resource allocation that optimizing the trade-off between travel cost and user experience. The new proposed system considers social network communities which helps improve the quality of the sharing service. Through map planning, we concentrate on the city blocks' connection with routs to make an efficient booking service. After that, the passengers within close communities are recommended by the proposed SONETS algorithm. In the experiment, we make an application and apply it well in the real environment. The results of large-scale simulation also show that our algorithm can provide cost efficiency, fairness, and humanized solutions for potential users, which leads to a higher quality of user experience.

6.1.3 Efficiency-oriented maintenance service using AI prediction

To improve the sustainability issues, this paper also studies the maintenance of pedelec system based on big data analytics, which meets the growing maintenance demand of new types. We have first studied the real-world urban data sets from all collect-able dimensions to explore the relations of different ingredients on the usage of sharing bikes. After that, a new hybrid predictive model to digitize the useful information in various fields and predict the future trend of bike usage has been developed. Finally, using the prediction outputs, a heuristic algorithm called intelligent share bike maintenance optimization algorithm (IMOA) has been proposed to solve the maintenance problem for replenishment trucks. Through simulations, we have verified our proposal from all angles and validate that the accuracy of our prediction model is high enough and IMOA solves the maintenance problem efficiently.

6.2 Future Work

In the future, we intend to extend our work in three dimensions. First, not limited to autonomous vehicles on the road, terahertz band can be widely applied to all mobile nanocell networks, such as delivery drones or UAVs (Unmanned Aerial Vehicle). Different kind of 'vehicles' has very different ability to adapt communication environments. We plan to transfer our technology to all kinds of vehicles. Next, humanized solutions for potential users, which leads to a higher quality of user experience, need a more solid measurements to show a specific factor for detail comparison. Then, we can arrange passengers more precisely. At last, the proposed system is feasible for NYC. However, the different landform and weather condition could significantly influence the bike usage and changes the inner connections between various datasets. We plan to develop a universal model to apply in different cites.

Meanwhile, the efficiency and performance bound of the proposed algorithm have not been thoroughly investigated. We intend to use more artificial intelligence technologies to improve the performance of the whole system.

References

[1] Abagnale, C., Cardone, M., Iodice, P., Strano, S., Terzo, M., and Vorraro, G. (2015). Power requirements and environmental impact of a pedelec. a case study based on real-life applications. *Environmental Impact Assessment Review*, 53:1 – 7.

[2] Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295 – 303.

[3] Akyildiz, I. F., Jornet, J. M., and Han, C. (2014). Terahertz band: Next frontier for wireless communications. *Physical Communication*, 12:16–32.

[4] Almannaa, M. H., Elhenawy, M., Ghanem, A., Ashqar, H. I., and Rakha, H. A. (2017). Network-wide bike availability clustering using the college admission algorithm: A case study of san francisco bay area. In *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 580–585.

[5] Caggiani, L., Camporeale, R., and Ottomanelli, M. (2017). A real time multi-objective cyclists route choice model for a bike-sharing mobile application. In *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 645–650.

[6] Chard, K., Bubendorfer, K., Caton, S., and Rana, O. F. (2012). Social cloud computing: A vision for socially motivated resource sharing. *IEEE Transactions on Services Computing*, 5(4):551–563.

[7] Chen, D. (2017). Solving a new type of tsp using genetic algorithm. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 3333–3339.

[8] Chen, P.-Y., Liu, J.-W., and Chen, W.-T. (2010). A fuel-saving and pollution-reducing dynamic taxi-sharing protocol in vanets. In *Proceedings of the 72nd IEEE Vehicular Technology Conference Fall (VTC 2010-Fall)*, pages 1–5.

[9] Chen, Q., Liu, M., and Liu, X. (2018). Bike fleet allocation models for repositioning in bike-sharing systems. *IEEE Intelligent Transportation Systems Magazine*, 10(1):19–29.

[10] Cheng, T. and Wicks, T. (2014). Event detection using twitter: A spatio-temporal approach. *PLoS ONE*, 9(6):e97807.

[11] Chusseau, L., Lampin, J., Bollaert, S., Duvillaret, L., and Mangeney, J. (2005). Thz active devices and applications: a survey of recent researches. In *Microwave Conference, 2005 European*, volume 1, pages 4–pp. IEEE.

[12] Cui, X., Gulliver, T. A., Li, J., and Zhang, H. (2016). Vehicle positioning using 5g millimeter-wave systems. *IEEE Access*, 4:6964–6973.

[13] Deshpande, P., Kodeswaran, P. A., Banerjee, N., Nanavati, A. A., Chhabra, D., and Kapoor, S. (2015). M4m: A model for enabling social network based sharing in the internet of things. In *Proceedings of the 7th International Conference on Communication Systems and Networks (COMSNETS 2015)*, pages 1–8.

[14] Dong, M., Liu, X., Qian, Z., Liu, A., and Wang, T. (2015). Qoe-ensured price competition model for emerging mobile networks. *Wireless Communications, IEEE*, 22(4):50–57.

[15] Dong, M., Ota, K., and Sakai, M. (2013). A novel information dissemination system for vehicle-to-rsu communication networks. In *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 918–919.

[16] d'Orey, P. M., Fernandes, R., and Ferreira, M. (2012). Empirical evaluation of a dynamic and distributed taxi-sharing system. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 140–146.

[17] Elloumi, W., Baklouti, N., Abraham, A., and Alimi, A. M. (2013). Hybridization of fuzzy pso and fuzzy aco applied to tsp. In *13th International Conference on Hybrid Intelligent Systems (HIS 2013)*, pages 105–110.

[18] Ezirim, K. and Jain, S. (2015). Taxi-cab cloud architecture to offload data traffic from cellular networks. In *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6.

[19] Fu, Y., Sun, D., Wang, Y., Feng, L., and Zhao, W. (2017). Multi-level load forecasting system based on power grid planning platform with integrated information. In *2017 Chinese Automation Congress (CAC)*, pages 933–938.

[20] Gong, D. and Ruan, X. (2004). A hybrid approach of ga and aco for tsp. In *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788)*, volume 3, pages 2068–2072 Vol.3.

[21] Guo, L., Dong, M., Ota, K., Li, Q., Ye, T., Wu, J., and Li, J. (2017). A secure mechanism for big data collection in large scale internet of vehicle. *IEEE Internet of Things Journal*, 4(2):601–610.

[22] Guo, S., Dong, M., and Guo, M. (2008). Performance analysis of heuristic algorithms for lifetime-aware directional multicasting in wireless ad hoc networks. In *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1311–1316.

[23] Hoang, M. X., Zheng, Y., and Singh, A. K. (2016). Fccf: Forecasting citywide crowd flows based on big data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '16, pages 6:1–6:10, New York, NY, USA. ACM.

[24] Honda, K., Nagata, Y., and Ono, I. (2013). A parallel genetic algorithm with edge assembly crossover for 100,000-city scale tsps. In *2013 IEEE Congress on Evolutionary Computation*, pages 1278–1285.

[25] Hosni, H. E., Farhat, N., Nimer, R., Alawieh, N., Masri, C. E., Saroufim, M., Artail, H., and Naoum-Sawaya, J. (2012). An optimization-based approach for passenger to shared taxi allocation. In *Proceedings of the 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2012)*, pages 1–7.

[26] Huang, Q. and Yuan, F. (2010). The linkage mechanisms between gas price and the taxi fare in china. In *Management and Service Science (MASS), 2010 International Conference on*, pages 1–4.

[27] Jia, J. and Zhang, Q. (2013). Rendezvous protocols based on message passing in cognitive radio networks. *IEEE Transactions on Wireless Communications*, 12(11):5594–5606.

[28] Jornet, J. M. and Akyildiz, I. F. (2011). Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the terahertz band. *IEEE Transactions on Wireless Communications*, 10(10):3211–3221.

[29] Khan, A., Correa, O., Tanin, E., Kulik, L., and Ramamohanarao, K. (2017). Ride-sharing is about agreeing on a destination. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 6. ACM.

[30] Kim, J.-T., Lee, J.-H., Kim, S., and Kim, I. K. (2013). Social contents sharing model and system based on user location and social network. In *Proceedings of the Third International Conference on Consumer Electronics Berlin (ICCEBerlin 2013)*, pages 314–318.

[31] Kleine-Ostmann, T. and Nagatsuma, T. (2011). A review on terahertz communications research. *Journal of Infrared, Millimeter, and Terahertz Waves*, 32(2):143–171.

[32] Kong, L., Khan, M. K., Wu, F., Chen, G., and Zeng, P. (2017). Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles: Overview, design, and challenges. *IEEE Communications Magazine*, 55(1):62–68.

[33] Kong, X., Xu, Z., Shen, G., Wang, J., Yang, Q., and Zhang, B. (2016). Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems*, 61:97–107.

[34] Kürner, T. (2012). Towards future thz communications systems. *Terahertz Science and Technology*, 5(1):11–17.

[35] Li, H., Ota, K., Dong, M., and Guo, M. (2017a). Mobile crowdsensing in software defined opportunistic networks. *IEEE Communications Magazine*, 55(6):140–145.

[36] Li, N., Oyler, D. W., Zhang, M., Yildiz, Y., Kolmanovsky, I., and Girard, A. R. (2017b). Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Transactions on Control Systems Technology*, PP(99):1–16.

[37] Li, Y., Gao, J., Lee, P. P. C., Su, L., He, C., He, C., Yang, F., and Fan, W. (2017c). A weighted crowdsourcing approach for network quality measurement in cellular data networks. *IEEE Transactions on Mobile Computing*, 16(2):300–313.

[38] Liu, C. T. and Hsu, R. C. (2015). A fuzzy q-learning based assisted power management method for comfortable riding of pedelec. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 580–585.

[39] Liu, Z., Dong, M., Zhang, B., Ji, Y., and Tanaka, Y. (2016a). Rmv: Real-time multi-view video streaming in highway vehicle ad-hoc networks (vanets). In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

[40] Liu, Z., Dong, M., Zhang, B., Ji, Y., and Tanaka, Y. (2016b). Rmv: Real-time multi-view video streaming in highway vehicle ad-hoc networks (vanets). In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

[41] Lomotey, R. K. and Deters, R. (2014). Architectural designs from mobile cloud computing to ubiquitous cloud computing - survey. pages 418–425.

[42] Lv, F., Zhu, H., Chang, S., and Dong, M. (2017). Synthesizing vehicle-to-vehicle communication trace for vanet research. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–3.

[43] Ma, S., Zheng, Y., and O., W. (2015). Real-time city-scale taxi ridesharing. *IEEE Transactions on Knowledge and Data Engineering*, 27(7):1782–1795.

[44] Minakov, I., Passerone, R., and Rossi, M. (2017). Design and energy optimization of a multifunctional iot solution for connected bikes. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6.

[45] Mohammed, M. M. A., He, C., and Armstrong, J. (2017). Performance analysis of aco-ofdm and dco-ofdm using bit and power loading in frequency selective optical wireless channels. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5.

[46] Nguyen, A. T., Li, B., Welzl, M., and F., E. (2011). Stir: Spontaneous social peer-to-peer streaming. In *Proceedings of IEEE Conference on The Computer Communications Workshops (INFOCOM WKSHPS 2011)*, pages 816–821.

[47] Ota, K., Dong, M., Chang, S., and Zhu, H. (2014). Mmcd: Max-throughput and min-delay cooperative downloading for drive-thru internet systems. In *Communications (ICC), 2014 IEEE International Conference on*, pages 83–87.

[48] Ota, K., Dong, M., Chang, S., and Zhu, H. (2015a). Mmcd: Cooperative downloading for highway vanets. *Emerging Topics in Computing, IEEE Transactions on*, 3(1):34–43.

[49] Ota, K., Dong, M., Chang, S., and Zhu, H. (2015b). Mmcd: Cooperative downloading for highway vanets. *IEEE Transactions on Emerging Topics in Computing*, 3(1):34–43.

[50] Ota, K., Dong, M., Zhu, H., Chang, S., and Shen, X. (2011). Traffic information prediction in urban vehicular networks: A correlation based approach. In *2011 IEEE Wireless Communications and Networking Conference*, pages 1021–1025.

[51] Pan, B., Zheng, Y., Wilkie, D., and Shahabi, C. (2013). Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, pages 344–353, New York, NY, USA. ACM.

[52] Paul, F. and Bogenberger, K. (2014). Evaluation-method for a station based urban-pedelec sharing system. *Transportation Research Procedia*, 4:482 – 493. Sustainable Mobility in Metropolitan Regions. mobil.TUM 2014. International Scientific Conference on Mobility and Transport. Conference Proceedings.

[53] Torres, S., Lalanne, F., del Canto, G., Morales, F., Bustos-Jimenez, J., and Reyes, P. (2015). Becity: sensing and sensibility on urban cycling for smarter cities. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–4.

[54] Vegni, A. M. and Loscri, V. (2015). A survey on vehicular social networks. *IEEE Communications Surveys Tutorials*, PP(99):1–1.

[55] Wang, X., Zhou, Z., Yang, Z., Liu, Y., and Peng, C. (2017). Spatio-temporal analysis and prediction of cellular traffic in metropolis. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pages 1–10.

[56] Wang, Y., Du, B., Rong, Q., and Lin, X. (2016). Travel patterns analysis of urban residents using automated fare collection system. *Chinese Journal of Electronics*, 25(1):40–47.

[57] Wu, J., Dong, M., Ota, K., Li, J., and Pei, B. (2014). A fine-grained cross-domain access control mechanism for social internet of things. In *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, pages 666–671.

[58] Yao, X., Shen, X., Wang, L., and He, T. (2017). Hybrid bicycle allocation for usage load balancing and lifetime optimization in bike-sharing systems. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pages 112–117.

[59] Yu, X. and Fan, C. (2014). Ecological analysis of smart city based on the numerical simulation technology. In *2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications*, pages 438–441.

[60] Yu, X., Pan, A., Tang, L.-A., Li, Z., and Han, J. (2011). Geo-friends recommendation in gps-based cyber-physical social network. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 361–368.

[61] Zhang, S. (2014). Influence of relationship strengths to network structures in social network. In *Communications and Information Technologies (ISCIT), 2014 14th International Symposium on*, pages 279–283.

[62] Zhang, S., Ma, Q., Zhang, Y., Liu, K., Zhu, T., and Liu, Y. (2015). Qa-share: Towards efficient qos-aware dispatching approach for urban taxi-sharing. In *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pages 533–541.

[63] Zheng, Y., Capra, L., Wolfson, O., and Yang, H. (2014). Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38.

[64] Zhou, Z., Dong, M., Ota, K., Shi, R., Liu, Z., and Takuro., S. (2015). Game-theoretic approach to energy-efficient resource allocation in device-to-device underlay communications. *Communications, IET*, 9(3):375–385.

[65] Zhou, Z., Ota, K., Dong, M., and Xu, C. (2017). Energy-efficient matching for resource allocation in d2d enabled cellular networks. *IEEE Transactions on Vehicular Technology*, 66(6):5256–5268.

Publications

Journals

1. Chaofeng Zhang, Mianxiong Dong, Kaoru Ota, Jian Qiu, Minyi Guo, "Social Taxi Sharing: A Cyber-Physical Approach for Efficient Urban Transportation Service," ACM Transactions on Cyber-Physical Systems (TCPS), In Press.

2. Chaofeng Zhang, Kaoru Ota, Juncheng Jia, Mianxiong Dong, "Breaking the Blockage for Big Data Transmission: Gigabit Road Communication in Autonomous Vehicles," IEEE Communications Magazine, vol. 56, no. 6, pp. 152-157, June 2018.

3. Chaofeng Zhang, Mianxiong Dong, Kaoru Ota, Minyi Guo, "Social Network Optimized Taxi Sharing Service for Smart Cities," IEEE IT Professional, vol. 18, no. 4, pp. 34-40, July-Aug. 2016.

Proceeding of International Conference

1. Chaofeng Zhang, Kaoru Ota, Mianxiong Dong, "Cooperative Positioning Optimization in Mobile Social Networks," in Proceedings of IEEE 84th Vehicular Technology Conference, 2016.

2. Hidekazu Suzuki, Mianxiong Dong, He Li, Kaoru Ota, Chaofeng Zhang, "Overlay Optimization for Cost Efficient P2P Streaming Service", Joint 8th International Conference on Soft Computing and Intelligent Systems and 17th International Symposium on Advanced Intelligent Systems (SCIS & ISIS2016), Sapporo, Hokkaido, Japan, August 25-28, 2016.

Under Review

1. Chaofeng Zhang, Mianxiong Dong, Tom H. Luan and Kaoru Ota, "Battery Maintenance of Pedelec Sharing System: Big Data based Usage Prediction and Replenishment Scheduling," IEEE Transactions on Network Science and Engineering, Minor Revision.