Technische Universität München

Chair of Transportation Systems Engineering

Master's Thesis

An optimization model for reservation based autonomous car sharing routing problem

Shun Su

supervised by

M.Sc. Emmanouil CHANIOTAKIS
Univ.-Prof.Dr. Constantinos ANTONIOU

May 13, 2018

Abstract

Share mobility is starting to gain popularity since recent years. It represents a transportation mode which contributes to reducing city traffic, congestion, and pollution problems. The new emerging transportation mode, shared autonomous vehicles, would help to overcome the limitations of the current vehicle sharing systems since it combines features of on-demand and short-term rentals with self-driving capabilities.

The thesis proposed an optimization model for reservation based car sharing (RACS) system by applying the concept of autonomous vehicle sharing chain (AVS Chain). Firstly, the RACS system and its routing optimization procedure based on the AVS Chain concept were defined. Then, the RACS system was formulated in terms of a mixed integer programming (MIP) problem. Since the problem denotes as a combinatorial optimization which is NP hardness, in order to obtain feasible solutions, meta-heuristic method of Tabu search and in combination with K-Means or K-Medoids clustering algorithms were applied. After that, the proposed model was validated by testing in different networks with small to large scale of real life travel demands. Furthermore, the performance and feasibility of the algorithms were evaluated.

The proposed AVS Chain model aims to minimize the AV travel time/cost and customer waiting time, taking both transportation systems efficiency and passengers' concern into account. It can be applied to transport networks of cities with small to large travel demands.

Keywords: Autonomous Vehicle(AV), Routing optimization, Reservation based car sharing (RACS), Mixed integer programming(MIP), Tabu search, K-Medoids, K-Means.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Prof. Dr. Constantinos Antoniou and Emmanouil Chaniotakis of the Chair of Transportation Systems Engineering at the Technical University of Munich. They always inspired me and provided me great support and guidance throughout my thesis. Also, I would like to give special thanks to my supervisor Emmanouil Chaniotakis who introduced me to this exciting research topic and continually provided me valuable advice and feedback with enormous patience and enthusiasm. You both taught me a great deal about both scientific research and life in general.

Furthermore, my sincere thanks also go to Dr. Tao Ma, Moeid Qurashi, and David Durán Rodas who always took their time to share their knowledge and discuss with me. Their assistance helped me to overcome several difficulties in the modeling and programming process.

Last but not least, I would like to thank my families and friends who have encouraged and supported me throughout my master's study.

Declaration

I hereby declare that this thesis is an outcome of my own efforts and has not been published anywhere else before and not used in any other examination. Also, to mention that the materials and methods used and quoted in this thesis has been properly referenced and acknowledged.

Place and dated:____________________ Signature:__________________

Contents

List of Figures

List of Tables

List of Abbreviations

AV	Autonomous Vehicle
AVs	Autonomous Vehicles
AVS	Autonomous Vehicle Sharing
AVS Chain	Autonomous Vehicle Sharing Chain
RACS	Reservation based Autonomous Car Sharing
DARP	Dail A Ride Problem
DARPs	Dail A Ride Problems
KMN-Tabu	K-Means Tabu
KMD-Tabu	K-Medoids Tabu
ICT	Information and Communication Technologies
SAE	Society of Automobile Engineers
SDARP	Static Dial-a-Ride Problem
DDARP	Dynamic Dial-a-Ride Problem
MIP	Mixed Integer Programming
PAM	Partitioning Around medoids algorithm
B2C	Business-to-Customer
DRT	Demand Responsive Transport
GAs	Genetic Algorithms
LNS	Large Neighborhood Search
P2P	peer-to-peer
SA	Simulated Annealing
TNC	Transport Network Company
VRP	Vehicle Routing Problem
TDVRPTW	Time dependent VRP with time windows

Chapter 1

Introduction

1.1 Background

Since recent years, the worldwide population growth and increasing urbanization have led to rapid expanding of urban areas, in which the travel demand raised significantly. The insufficient supply of transportation infrastructure and services in the urban area has accelerated the increasing use of private vehicles. Nowadays, private vehicle mobility has become the dominated transport mode in a vast majority of the cities around the world. According to Viegas et al. [2016], the private car trip accounts for 40 to 83 percent of the total trip demand in the cities.

Although private vehicles have many advantages, such as flexibility, comfort, and good availability, they are also the primary contributor to traffic congestion and environmental pollution. Moreover, the transport efficiency of private vehicles is considered as low, since most vehicles have a quite low daily usage rate and require parking land at the idle time. Litman [2017] presented an example of private vehicle usage in Austin (US) and indicated that the private vehicles remained unused for 23 hours per day on average and increased the parking land, which was equal to 25 percent of the total urban area. To solve this problem, a variety of transportation management measures have been proposed by traffic authorities. Their approaches can be categorized into the following groups: a) shifting demand to more sustainable transportation modes; b) demand reduction measures; c) advanced and intelligent management of traffic; d) policy and laws.

Shared mobility emerges as a sustainable transportation mode which could significantly reduce the ownership of vehicles, average transport costs, greenhouse gas emissions, and congestion [Shaheen et al., 2015]. Given these potential benefits, shared mobility services including car sharing, ride sharing, and bike sharing are today approved and encouraged by traffic authorities of different countries. It is forecasted that shared mobility will keep growing rapidly in the next few years and become an important transportation mode which compensates for public transport and taxi services [Corwin et al., 2015]. However, the current shared mobility services are still facing some challenges, such as limited availability of vehicles and service levels, relatively high operation costs resulting from the relocation and re-balancing of the shared vehicles [Le Vine et al., 2014]. These challenges are expected to be overcome by introducing autonomous vehicles (AVs) in the shared mobility business.

Autonomous vehicles are predicted to be increasingly utilized in different fields and will arouse a revolution in the transportation systems [Ambühl et al., 2016]. The shared autonomous vehicles, for example, could complete the relocation and re-balancing by self-driving and thus reduce the operation costs and cut the customer waiting time significantly. It is expected that shared AVs could be an important part of the future transportation systems, which offers low-cost, less-stressful, more safety services, and integrates with other transportation modes to provide on-demand and seamless journeys for passengers [Krueger et al., 2016].

The positive impacts of applying AVs in the current vehicle sharing systems have been proved by large number of studies [Stocker and Shaheen, 2017, Dia and Javanshour, 2017, Litman, 2018]. In addition, to operate the Shared AVs more properly and efficiently, various operation strategies and optimization methods are proposed and discussed. Among the different aspects of shard AVs operation, the AV routing problem is the most critical since it determines not only the system capability of passenger transportation, but also the vehicle utilization efficiency. This thesis commits to contribute to the autonomous vehicle sharing routing optimization problems.

1.2 Research problem statement

The research problem was initiated by the chair of Transportation Systems Engineering at the Technical University of Munich. The proposal was concerned about developing an optimized shared autonomous vehicles model based on the sharing chain concept, taking the diverse travel demands into account. It was expected that the new model would contribute to the efficient operation of urban transportation systems and prepare itself to fulfill the growing travel demands.

A closer look at the existing literature reveals that few literature studied the reservation based operation strategies for shared AVs, which, however, could have significant potential benefits on improving mobility and sustainability of current transportation system [Ma et al., 2017]. Furthermore, research on optimization problem for shared AVs from perspectives of both systems and passengers are quite scarce. Most research only focus on system capacity and operation level. In addition, few proposed models were solved based on the real-life large scale of travel demands. This thesis seeks to fill this research gap by investing reservation based autonomous car sharing (RACS) system and develop an optimization routing model which reflects both system and passengers benefits and is applicable for large scale of travel demands.

In this thesis, the scope of research is limited in routing within a single planning horizon, which assumes that all traffic information and travel requests/demands are known in advance. This period of time is regulated by the system and static plan of the routes. In addition, this thesis focuses on car sharing system, thus ride sharing and bike sharing are not considered.

1.3 Objective and research question

The key issue related to the successful operation of RACS system is AV routing optimization and there are two major problems need to be focused on: a) formulation the of the autonomous vehicle sharing chain model which indicates the optimization process of routing and b) utilization of novel solution approaches. The main objective is to establish an optimization model for RACS system by applying the concept of autonomous vehicle sharing (AVS) chains. Moreover, utilize appropriate algorithms to solve different size problems according to the real life demand, then evaluate each algorithm performance.

Main Research questions:
Given the aforementioned need the following two main research questions are formulated:

1. How to optimize the reservation based autonomous vehicle sharing system by applying AVS Chain model concept and build the optimization model to minimize the cost of vehicle travel time and customer waiting time?
2. How to solve the model with appropriate algorithms with different scale of demand problems ?

The research questions to be answered introduces a wide spectrum of secondary research questions:

1. What is autonomous vehicle sharing system and how it could classified based on business and operation mode?

2. What is the reservation based autonomous car sharing system and how it perform?
3. How to model a reservation based autonomous car sharing system?
4. What is the state of art in autonomous vehicle routing for reservation based autonomous car sharing system?
5. How to define the autonomous car sharing routing problem?
6. How to define a "AVS Chain" concept?
7. What methodologies could be applied to the modelling and what are the decision variables, objective functions and constraints?
8. How to solve the model with appropriate algorithms with different scale of demand problems?
9. How are the algorithms applied in the solution process?
10. What are the performance and case study results of the algorithms?

1.4 Contributions

This thesis contributions lay on both modelling and algorithm methodological and practical level:

1. Propose autonomous vehicle sharing (AVS) chain concept to effectively represent the reservation based car sharing (RACS) routing optimization process.
2. Consider both system and customer objectives into the problem modelling procedure, which include minimizing the total travel cost and customer waiting time, as well as formulate the problem as a mixed integer programming optimization.
3. Propose, Formulate and Apply clustering methods of K-Means and K-Medoids with meta-heuristic Tabu search algorithm into the solution procedure.
4. Test the three algorithm including Tabu, K-Means-Tabu (KMN-Tabu) and K-Medoids-Tabu (KMD-Tabu) with small and large scale cases. By comparing their performance based on solution quality, stability and computation cost, conclude the most suitable algorithm for each scale problems in real life.

1.5 Research Framework

This thesis was performed following the following research framework (Figure 1.1).

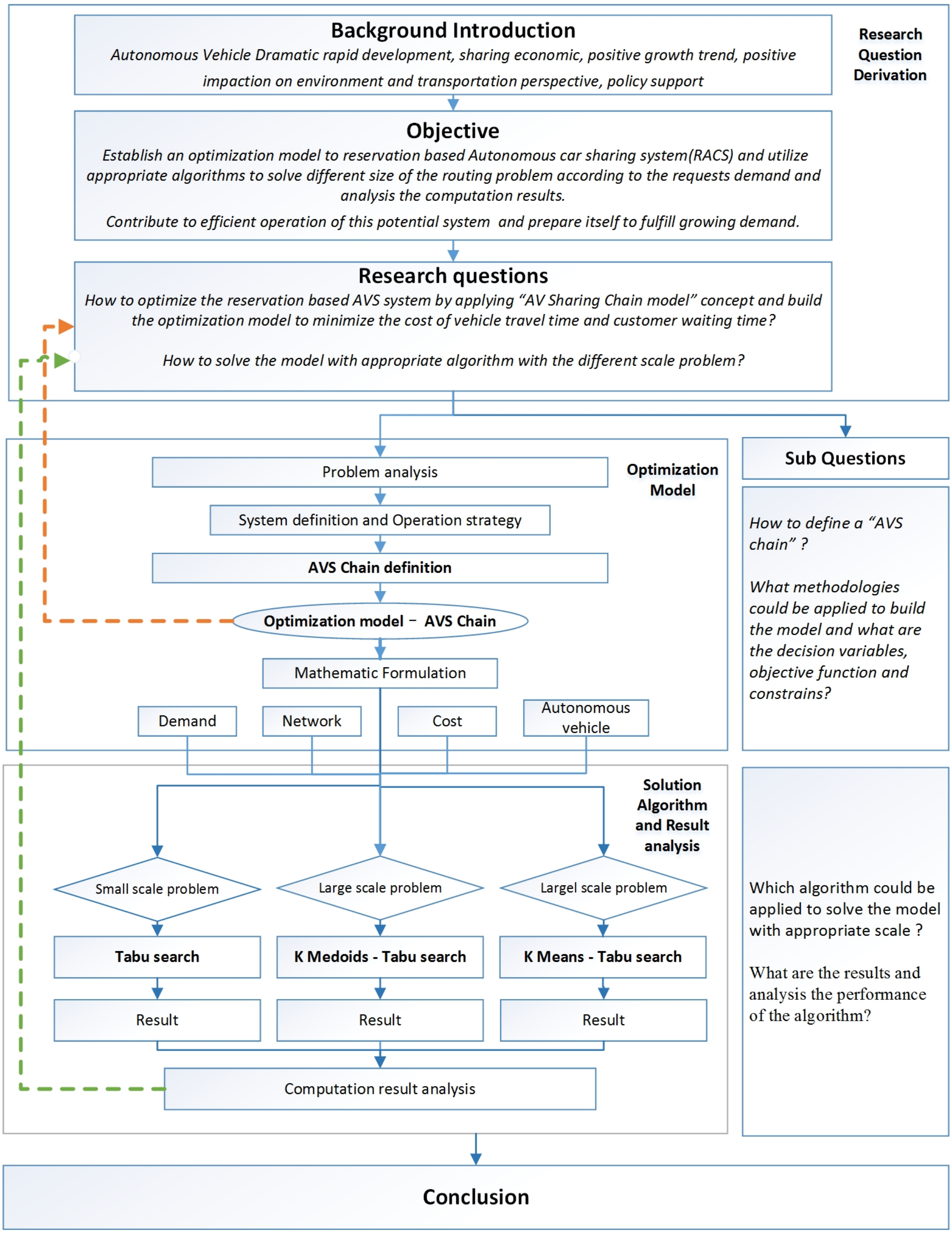

Figure 1.1: Research flow

1.6 Report Structure

This thesis is structured as follows:

- Chapter 1 introduces the background of the problem and raise the objective of research questions, defines the research scope and research framework.
- Chapter 2 represents the most relevant subjects for this thesis, from the definition of vehicle sharing system, autonomous vehicles sharing, to the optimization modelling methods of RACS system and the related solution methods..
- Chapter 3 presents the definition of the routing optimization procedure of the RACS system based on AVS chain concept. Further, the model was formulated.
- Chapter 4 outlines the solution methods and the set of algorithms which are applied to solve the RACS model.
- Chapter 5 represents the solution procedure of the applied algorithms. Further, the function of each algorithm was explained step by step.
- Chapter 6 presents the computation results of the model in three different networks and demand settings, applying different solution algorithms. Further, the feasibility and performance of algorithms are discussed.
- Chapter 7 sums up the key findings in this thesis and proposes ideas and recommendations for further research.

Chapter 2

Literature review

In this section, related work is reviewed and analyzed in order to acquire a deeper understanding of the problem in hand, and find the most appropriate methodologies to identify gaps in the literature and guide towards the development of methods that answer the research questions. In this chapter the following research secondary questions are addressed.
1. What is autonomous vehicle sharing system and how it could classified based on business and operation mode?
2. What is the reservation based autonomous car sharing system and how it perform?
3. How to model a reservation based autonomous car sharing system ?
4. What is the state of art in autonomous vehicle routing for reservation based autonomous car sharing system?

2.1 Vehicle sharing system

2.1.1 Shared mobility

Shared mobility refers to the shared use of a vehicle, bicycle, or even rides that enables users to have short-term access to transportation services on an on-demand basis [Shaheen et al., 2015]. As the public perception of shared goods has changed considerably in the last years, the shared mobility is now beginning to gain widespread popularity. Shared mobility promise to reduce inner-city traffic, congestion, and pollution problems, which are particularly interested for the cities that are struggling with population growth and increasing density [Cohen and Kietzmann, 2014]. In 2016, the global shared mobility market was nearly 54 billion [Consumer and Boomers, 2017]. In respect of the strong customer demand, the shared mobility market is expected to experience impressive annual growth rates in the future [Corwin et al., 2015]. Driven by the improved automation, information and communication technologies, numerous new shared mobility concepts are emerging, which enable passengers to use different transportation modes in more efficient and sustainable ways. Generally, the current existing shared mobility business models can be classified into car sharing, ride sharing, and bike sharing [Cohen and Kietzmann, 2014]. The service providers for the shared-used vehicles are: car sharing providers, such as car2go and DriveNow, who owe or lease a fleet of vehicles and connect the passengers to cars; and ridesourcing providers, or called transport network companies (TNCs), such as Lyft and Uber, who connect the drivers to passengers (taxi, ride sharing) [Consumer and Boomers, 2017].

2.1.2 Car sharing business models

Car sharing is a model of short-term car rental, which allows customers to use the vehicle service for shorter periods of time, usually on a per hour basis. Passengers who registered as customers of the car sharing companies can request vehicles through hotline, website, or mobile Apps, and get access to the shared-used vehicles and drive by themselves. The customers of car sharing can gain benefits of shared vehicle utilization without ownership, fuel, and parking costs, only response to the fees based on their traveling distances or utilization time period [Shaheen et al., 2015]. Car sharing is particularly attractive to customers, who only need to use vehicles occasionally or for shorter distance trips in urban areas, where personal car ownership can be challenging. In recent years, car sharing has experienced continuously double-digital growth [Consumer and Boomers, 2017]. Aiming to fulfill the diverse range of customer experience and at differentiated price points, a broad range of car sharing business models are emerged. These models can be classified into round trip, point-to-point free-floating, point-to-point station-based, and peer-to-peer (P2P) [Le Vine et al., 2014]. While round trip models require customer to return the vehicle to the same place where it was accessed, point-to-point (also called one-way) models enable one-way journeys within a specified geographic zone and allow customers to leave the vehicle parked on the street near the destination. Compared with round trip models, point-to-point models allow customers to use the vehicles spontaneously, or reserve the vehicles only several minutes in advance [Le Vine et al., 2014]. However, the logistics of point-to-point models, especially with free-floating systems are most challenging to manage, since the car sharing providers have to reallocate the vehicles to the dedicated places in time. Some of the point-to-point car sharing are station-based. They require customers to pick up the vehicles from one parking station and return them to another, where fixed infrastructure such as charging points for electric vehicles can be located. Differ from the above described Business-to-Customer (B2C) car sharing models, peer-to-peer (P2P) models are operated by companies who typically do not own any of the shared-used vehicles, but supervise transaction among individual owners and renters and provide the necessary platform and resources needed for the exchange [Stocker and Shaheen, 2017].

2.1.3 Ride sharing business models

Ride sharing is another important shared mobility model in which drivers and riders with similar origins and/or destinations share the rides. It would provide significant societal and environmental benefits by reducing the number of vehicles and improving the utilization of available sear capacity. Since several riders share a vehicle, this model usually has a benefit of lower costs compared with car sharing. Nowadays, the optimization technology which matches drivers and riders effectively and efficiently is the key for ride sharing business models [Masoud and Jayakrishnan, 2017]. Ride sharing can be further categorized into carpooling, flexible carpooling, vanpooling, and P2P ride sharing [Agatz et al., 2012].

2.2 Potential autonomous vehicle sharing design

2.2.1 Autonomous vehicle and sharing definition

2.2.1.1 Autonomous vehicle

An autonomous vehicle is that one vehicle with partially or fully driver-less function and can be maneuvered with reduced or no human intervention [Litman, 2017]. Based on the advanced degree of the self-driving system in the vehicle, five levels of automation for on-road vehicles have been defined in the standard J3016 composed by Society of Automobile Engineers (SAE) International. According to this standard, the Level 4 and Level 5 allow driver-less operation, in which the Level 5 has the highest level of automation and could possess fully autonomous in all circumstances without any human intervention [Litman, 2018]. Herein, none but level 5 and its related features are considered in this thesis.

The autonomous vehicles with Level 5 should be equipped with different advanced technologies and systems. They are such as advanced sensors (optical, infrared, radar, laser, etc.); automation controls system for steering, braking, and signals; navigation GPS based on the high-quality maps system; vehicle-to-vehicle communication system; information collection system; and testing and maintenance systems for critical components [Litman, 2017]. Further, AVs can be classified according to the capacity of passengers and types of power supplement, which are also essential features of AVs. There are normal AVs with 4-6 seats, mini autonomous shuttle or vans with 8-10 sears, and autonomous buses with 10-20 seats [Litman, 2017, Hörl, 2016, Alessandrini et al., 2015]. Besides, the type of power supplement, such as fuel or electric, should be considered in the development and operation of AVs [Kang et al., 2016]. In order to be aware of the public attitudes and willingness to utilize the AVs, Bansal et al. [2016] conducted a survey which shows positive results, especially for people among the group of without driving skills, younger generation and people with accident experiences.

In this thesis, the author defines autonomous vehicles as fully automated vehicles with level 5. The optimization model was thus established based on the assumption that all autonomous vehicles in the system are fully autonomous without the need of human drivers.

2.2.1.2 Autonomous vehicle sharing (AVS)

AVs are considered as a solution to the current transportation problems since they are safer, environment-friendly, and efficient [Litman, 2017]. One of the critical issues related to AV is maximizing the AVs capacity in current or future transportation systems. Among different potential choices, sharing models are most discussed [Masoud and Jayakrishnan, 2016].

AV sharing system is expected to become an important part of the future urban transportation system, which has positive impacts on transport convenience and safety, environment, and economics [Litman, 2017]. The business models of an AVS system can be described as follows: The AVS system will provide nearby or point-to-point on-demand mobility services with a fleet of AVs for passengers [Fagnant and Kockelman, 2014]. Passengers could contact the AVS system via designate methods to request a service, and the system will dispatch an AV to fulfill this request through a regulated operation process. Due to the features of driver-less and low cost, AVS system will be increasingly attractive to passengers and predicted to replace the personal vehicle travel demands partially [Litman, 2018].

The findings of the pertinent literature on the impacts of AVS with regards to the urban transportation system are summarized as follows:

1. Cost Reduction. Encourage shared ownership covering the high cost of purchasing vehicles to make AVs more affordable. Moreover, the driver-less feature of AVs will convenient the utilization and rent of AVs whenever they are unused by the owners [Litman, 2017].

2. Decrease the total number of vehicles needed for passengers. AVs can drive themselves to locations where travel demands arise. Based on the research of Fagnant and Kockelman [2013], in sharing system, AVs can provide five times capacity compared to non-shared vehicles with same fleet size, it will decrease the number of vehicles required for daily transport tasks.

3. Parking demand reduction. The increased utilization of AVs will generate less idle time and reduce the space for parking needs. Additionally, self-parking features of AVS will contribute to move the parking space outside the high-density area, and create more public areas for other utilization to gain more benefits on land use, especially for central business districts. The reduction of parking will also decrease the infrastructure maintenance cost for parking facilities [Zhang et al., 2015].

4. A higher degree of safety. According to the traffic accidents data by world bank [Peden et al., 2004], most of the accidents are caused by human error. With advanced control system and automated driving technologies in the AV sharing system, injury rates and fatal crashes caused by driver distraction would be tremendously reduced [Gruel and Stanford, 2016].

5. Reduction of environmental impacts. With the decreasing number of vehicles on roads, emissions and noise would be gradually reduced as well [Litman, 2018].

2.2.2 Autonomous vehicle sharing design

Since the AVS system is expected to be an important transportation mode in future, its characteristics and attributes must be designed for all kinds of passenger groups and travel purposes. In the previous studies, two major types of business models of AVS have been defined: autonomous car sharing and autonomous ride sharing [Zhang et al., 2015], In accordance with the expected goals of AVS system development aforementioned, various operation strategies have been proposed based on these two models.

2.2.2.1 Autonomous car sharing

Autonomous car sharing systems provide on-demand mobility which serves requests sequentially. Many studies have proposed their autonomous car sharing designs based on a door-to-door services feature. For instance, Ford [2012] investigated the autonomous taxi service for passenger delivery. In the system, relocation of AVs to higher demand area by self-driving are permitted. Spieser et al. [2014] proposed a system by modifying the traditional one-way based car sharing and transit into an autonomous taxi system. Meanwhile, Correia and van Arem [2016] proposed a public or private owned autonomous car sharing model as low-cost taxi system to provide dial-a-ride services to single or group passengers. By simulating the replacement of private vehicles with shared AVs in urban transportation, they explore the impacts of AVS system on traffic delays and parking. Similar research also has been conducted by Bischoff and Maciejewski [2016] who developed a simulation model to analyze different penetration rate scenarios with various size of autonomous taxi fleets to replace all private vehicles trips in the city of Berlin. By providing a real-time dispatching method, the result successfully proved that all private vehicles could be replaced by much less shared AVs for same demand.

2.2.2.2 Operation strategies of autonomous car sharing

Most of the previous designs assumed that autonomous car sharing would use the similar interactive pattern between system and customers through the well developed smart-phone APPs or ICT devices for demand ordering. Therefore, if any customers wish to use the AVS service, they have to make requests with following steps [Kang et al., 2016].

Process of utilize the AVS service a) Register in the system b) Make request via internet or APPs, then wait for AV pickup, c) AV arrive and execute the delivery d) drop off customer, AV relocate itself to next request. To encourage more passengers to use the AVS service and maintain the competitive advantages, different operation strategies need to be developed to satisfy the diverse trip purposes and customer preferences. For instance, one passenger wishes to use a vehicle for shopping immediately, and another wants to reserve a trip to the airport the next morning. Therefore, most of the strategies are separately considered with long term and short term requests, defining a framework that corresponds to static and dynamic optimization. The majority of studies focus on the dynamic mode since it is very common for these services to emerge within a short period of time [Dia and Javanshour, 2017, Hörl, 2017, Viegas et al., 2016]. On the static optimization problem, Ma et al. [2017] has proposed a reservation based AVS system,

in which customers can reserve trip requests ahead of time and the system operator will optimally assign the requests to routes. Moreover, Lamotte et al. [2017] also used the reservation based strategy for demand management and investigated the potential benefits with the road capacity allocation measure for shared AVs.
The comparison of two operation strategies are showed in the table 2.1 [Ma et al., 2017, Levin et al., 2016], where the difference could be find from customer and system activities. In this thesis, reservation based car sharing system is the main focus and more content concerning this system will be present afterwards.

Table 2.1: Autonomous car sharing system operation strategy comparison

Operation strategy	Customer	System
Static(reservation based) System operate based on planning time demand accepted	1.Customer makes reservation before the indicated planning time. 2. then arrives at the designated location and waits for the vehicle, when arrived, boarding. 3. When the vehicle arrives at its destination, leave system	1.Accept the requests within the system planning period. 2. System calculation and vehicle dispatching. Find best possible routes and assign the most appropriate type of vehicle to implement the task. 3.When all services have been provided then relocated itself to terminal station.
Dynamic (Real time) System operates based on real time requests accepted	1. Customer makes requests just before they need service. 2. Then arrives at the designated location and waits for the vehicle and boarding. 3. Arrives destination and drop off, customer leave the system.	1.System accept the service request. 2.Request dispatching and routing plan. 3.Service provided and waiting for next request or relocated itself to higher demand area.

2.2.2.3 Autonomous ride sharing

In contrast to autonomous car sharing, autonomous ride sharing allows the acceptance of requests while in service based with constrained capacity. It is usually designed to serve dynamic travel demand [Walker and Marchau, 2017, Fagnant and Kockelman, 2014]. From the perspective of resource utilization, it could promote the utilization ratio of empty seats in vehicles and lowers the fuel consumption and congestion, at least in comparison to non-ride sharing mobility [Litman, 2018]. Since the ride sharing service charges are relatively low, it is believed that average or low-income individuals would be the primary target population [Litman, 2018].

An efficient and effortless ride-sharing service could bring the shared AVs to its full potential [Litman, 2018]. Many studies in the pertinent literature has focused on problems that might arise during AVs deployment. For instance, Fagnant and Kockelman [2014] propose a ride sharing utilization of AVs. Customers could share one vehicle by reasonable scheduling the requested trips. Another dynamic ride sharing based new system called 'Automated taxis (ATs)' is proposed by Walker and Marchau [2017] for passenger transport. It provides pickup and drop off service based on the trip origin and destination while dynamically accepting requests. During the trip, AVs travel on optimal routes provided by the system's management center. Moreover, Kornhauser [2013] evaluate a simulation model of the AVS system named aTaxi system with two dynamic ride sharing implications for all private trips operating through New Jersey's network. In their model, aTaxi will wait for passengers boarding within a fixed amount of time before departing and provide ride sharing service to customers who have similar destinations. Dedicated stations are used rather than door-to-door service in the aTaxi system where passengers have to make a short distance walk to their destinations. Lam et al. [2015] designed a new AVs public transportation system, which provides ride sharing services with a fleet of AVs to accommodate requests of passengers. Similarly, Rigole [2014] applied a dynamic (real time) autonomous ride sharing service model in Stockholm, and proposed a simulation method to assess the benefits of this AVS system by replacing private vehicle trips of commuting. In the model, passenger waiting time, trip times and fleet size were considered as the key performance indicators.

2.2.2.4 Other types of autonomous vehicle sharing system

Besides the two major types of the AVS system, some researchers designed the AVS system with both autonomous ride sharing and car sharing function. Ford [2012] designed a shared autonomous Taxi system to operate with both business models according to temporal traffic demand. During non-peak hours, it operates as an autonomous car sharing system to provide higher quality service, while, in peak hours, the system switches the model and operates as a private rapid transit system by promoting ride sharing to fulfill a large portion of demand.

Apart from the business model difference, more advanced elements have also been considered in the system design, such as Kang et al. [2016] accounted for the AVs sharing system with electric vehicles and analyzed the relationships among vehicles, charging stations (CS), electric powertrain performance, and consumer demand. They also presented an optimization framework system design which integrated four subsystem problems as fleet size and scheduling, locations of charging stations, vehicle power, and service fees.

2.3 Reservation based autonomous car sharing (RACS) system

As aforementioned in section 2.2.2.2,reservation based autonomous car sharing system could be seen as one of the subsystems of autonomous car sharing which applies the static operation strategy.

This kind of system has been seldom studied until recently. Lamotte et al. [2017] investigated the scenarios of reservation based AVS system and its impact on congestion. Ma et al. [2017] proposed a reservation based AVS system, in which customers can reserve trip requests ahead of time and the system operator optimally routing for each shared AVs. They have tested the routing model with NewYork taxi data. Moreover, Pimenta et al. [2017] designed a small autonomous electrical vehicle system which operated as based on static Dial a Ride business.

Characteristics of the system

Based on RACS characteristics, the corresponding potential features [Ma et al., 2017] can be assumed as follows:

1. The system is operated by a management and control center.
2. A fleet of AVs in the system provide pickup and drop off services which will be shared by the customers sequentially based on their requests time in the network. Different types of vehicles with various capacity or additional functions may provide in the system.
3. Each AV can only fulfill one request at one time, without any disturbance from other pick and delivery requests.
4. Each request can take one or more passengers at the same time based on vehicle seats. If the passenger number exceeds the capacity, either the operator dispatches another type of vehicle which fits the requirement or customers split into subgroups and make several requests. The maximum number of passengers for each group will be regulated according to the different type of vehicles. The system will select the most appropriate vehicle automatically or plan for customers during the reservation process.

5. Customers need to make requests during the system regulated time period and specify the number of passengers, OD locations and departure time.
6. Concerning the departure time, the operator needs to take the tolerance of pickup delay into account and form a time window for each request as earliest and latest pickup time.
7. Operators need to estimate potential travel time on the planed traverse paths of each request and based on the predicted data to execute routing optimization.

Due to the reservation strategy of RACS system, the service may only fits certain group of customers with certain travel patterns or preference as the target customers:

- Customers with relatively fixed travel patterns, such as daily home to work trip.
- Customers who wish to enjoy privacy environment.
- Customers who have already arranged trip plans for a relatively long time before their preferred departure time.

2.4 RACS routing optimization modeling

In accordance with the characteristics mentioned above, RACS provide mobility services with pairwise location pickup and drop off. Therefore, modeling of this system operation can be generally described as a routing problem of dispatching shared AVs to requests. Modeling this problem should efficiently reflect the routing process and its components, constraints, and objectives.

Notwithstanding, compare to impact analysis, only a few literature refer to this problem. To name but a few, Levin [2017] has proposed a linear program for optimally dynamic traffic assignment and congestion awareness routing model, however without taking into account the time window constraints. Ma et al. [2017] designed a autonomous car sharing and reservation based system by applying a linear programming approach, and tried to solve the optimal routing problem with minimum travel cost in a designated planning horizon, where time window and other constraints are included. Meanwhile, Pimenta et al. [2017] proposed an integer linear programming formulation which aims to optimize a Dial a Ride system routing problem which involves small electric vehicles and objective of the minimum number of loading/unloading.

2.4.1 Model definition

To effectively reflect the system routing problem within a precise mathematical formulation in the modeling process, firstly, it is necessary to understand the related components, objective and constraints which give a clear definition and scale of the problem from the optimization perspective. Nevertheless, based on the real-life large demand situation, one of the main issues related to large size optimization problem is the increase of dimensions when moving towards a large number of requests. This is known as NP-hardness and related to combinatorial optimization (as the one included here). Therefore, the routing problem could be separately analyzed with steps by step.

2.4.1.1 Components

Autonomous car sharing routing is a systematic problem which constitutes five major components:

1. Demand model - Reserved requests matrix including variables of origin and destination, number of passenger and departure time for each request [Pimenta et al., 2017].
2. Network model - Road network in which the system provide services of mobility, it contains nodes and links as a graph [Levin et al., 2016].
3. Vehicles - a fleet of AVs represents the mobiles performing the request tasks.
4. Cost model - travel time or cost of each link or paths the shared car traversed by [Ma et al., 2017].
5. Routing optimization model - mathematical formulation which present the optimization processes including the request assignment and vehicle routing to achieve the system operational objectives.

2.4.1.2 Constraints

1. System capacity -all elements which related to capacity including number of AVs in the fleet, types of AV, depots (terminals), charging stations etc [Levin, 2017].
2. Business model - Provide autonomous car sharing mobility with pickup and drop-off service to customers in road network.
3. Operation strategy - reservation based.

4. Management and control -including vehicle routes plan, vehicle refueling/recharging, maintenance etc. Among them the vehicle routing is the most important issue which directly determines the successful operation of the system [Pimenta et al., 2017].

2.4.1.3 Objectives

Optimally routing the vehicles must based on a precise objective or multiple objectives, which will guide the optimization process to right direction. From different perspective of problem participants, objectives can be summarized as:

1. System : minimum total travel time of the fleet, total operation cost, maintenance cost [Ma et al., 2017], max profits etc [Ma et al., 2017].
2. Customer: minimum waiting time, max quality of service etc [Cordeau, 2006].
3. Municipal: minimum congestion time and lowest environment impacts.

During the modelling process, objectives could be separately or combined considered..

2.4.2 Vehicle Routing problem

Vehicle routing is a classic operational research question covering a wide variety of problems, which mainly works on logistics of transportation and management fields. The basic definition is routing a fleet of vehicles to provide the particular type of transport services for requests with passengers or goods [Toth and Vigo, 2002]. Since it represents a family of similar problems, the definition varies based on the problem complexity and additional constraints [Laporte et al., 2003]. Therefore, to clearly define our RACS routing model with the most appropriate type of vehicle routing problem is essential. For an extensive review, typical extension introduced to the basic vehicle routing problem definition could be seen in Table 2.2 [Correia and van Arem, 2016], and some of the extensions could be combined to form a new problem.

From these extended definitions, although there is some difference between the system component and constraints, RACS system problem still can be described as a capacitated VRP, formed by pickup and delivery of passengers with time windows, which is most relevant to the definition of Dial A Ride problem (DARP). After that, the variants and characteristic of DARP will be further discussed.

Since the RACS routing optimization is under a static condition, it can provide an upper bound of optimal solutions for different AVS systems, including ride sharing [Ma et al.,

Table 2.2: Types of Vehicle Routing Problem

Type of VRP	**Extension (Added Constraints)**
Classic VRP	A fleet of vehicles transport goods or passengers with the routes to fulfill its operation optimal objective.
Capacitated VRP	Limited capacity vehicles on goods or passengers in each trip
VRP with time windows	A mandatory or preferred time window for the pick-up and/or delivery of a good or passenger at a certain node
VRP with multiple trips	Vehicles can perform several trips starting at the depot.
Open VRP	The vehicle does not need to end its trip at the depot but instead it can park at the last client node.
Pick-up and delivery problem	Vehicles have to transport loads from origins to destinations
Dial-a-ride Problem (DARP) or demand responsive transport (DRT) problem	The pick-up and delivery problem for passengers, time window, maximum trip time of passenger etc.
The time dependent VRP with time windows (TDVRPTW)	Travel times varied according to time peroid in the network and time windows
Dynamic Vehicle Routing	Real-time availability of information for generating a change on the route or schedule of a vehicle

2017]. However, routing optimization severely depends on the data of travel time on each paths vehicle traversed. Hence, an efficient traffic situation predictive program is indispensable. Otherwise, the quality of solution would be seriously influenced by the data which is not precise enough.

2.4.3 Dial A Ride problem

Dial a ride problem (DARP) is defined upon a specific demand of customers in the road network, for which, each request comprises of a pickup at one location and drop off at another. The goal is to schedule a fleet of vehicles with the optimal sequence to satisfied all the requests. Standard DARP usually has one depot where the vehicles start and end their operation. The objective is to minimize total transportation costs taking into account the level of service provided to the passengers [Cordeau and Laporte, 2007].

From the modelling perspective, DARP is a combination of pickup and delivery vehicle routing problem with time window for passengers. It is different from the other Vehicle Routing Problems since it takes into account both system performance and customers satisfaction by minimizing the operation travel time cost and reducing a quantity related to the inconvenience of the passenger.

2.4.3.1 Objectives and constraints

Most of the DARP studies assume that all vehicles in a fleet are homogeneous, and only one terminal/depot is used for the whole system [Ho et al., 2018a, Karabuk, 2009, Qu and Bard, 2013]. However, to reflect the different reality situations in practice, based on the diverse need of customers, the fleet may be heterogeneous, such as normal and large capacity vehicle or vans are provided in the system for group customers [Braekers et al., 2014]. Meanwhile, in large scale networks, one depot may not be enough for the whole fleet's number of vehicles (for parking and maintenance). Conflicting objectives are also present in the literature. For example, some focus on determining the minimum cost of vehicle routes to fulfill all the demand requests [Luo and Schonfeld, 2007], nevertheless, in other contexts, the system aims to serve the maximum the requests with a fixed fleet size capacity with minimum violation of constraints [Cordeau and Laporte, 2007]. Herein, the cost may composite with fleet cost, travel time/cost and maintenance cost etc. The violation of constraints related to customers may count as the punishment cost [Cordeau and Laporte, 2007]. Some of the widely used objectives and constraint are concluded as following:

Objectives

- System perspective: total travel time of fleet minimum, minimum fleet size, minimum idle time of the vehicle, the maximum utilization rate of the vehicle.
- Customers perspective: minimum violation of ride time, the minimum waiting time for pickup.

  Most of the dial a ride problems DARPs incorporate several objectives aforementioned into the objective functions [Paquette et al., 2013, Lehuédé et al., 2014].

Constraints [Ho et al., 2018a]:

- Round trip: each trip of vehicle departure and terminate at the depot.
- Pairwise: each request has one pair of original and destination nodes, this pair of nodes need to be on the same route

- Precedence: the destination node must be visited after the original node for each request.
- Vehicle capacity: each type of vehicles has a limited capacity of seats.
- Time Window: a pick-up or drop-off time at specified time period must be given by customers as a time windows.
- Route duration: each route length of time should not exceed a limit settled by the system which also could see as planning horizon.
- Maximum ride time: customers' trip time on the vehicle should not exceed a specified maximum ride time.

2.4.3.2 Taxonomy of DARP

DARP is a set of optimization problems, which can be classified into static and dynamic, deterministic and stochastic according to the variations of the demand, traffic information awareness and the number of vehicle [Cordeau and Laporte, 2007].

Static and dynamic:

- Static DARP: The static Dial-a-Ride Problem (SDARP) assumes that all the travel demand are known ahead of time, routing of the vehicles have already been planned before the operation begun. Therefore, when it comes to operation, all routes for vehicle are fixed, it can be performed off-line. SDARP could also be applied to operation strategy, planning and long-term decision study. Various scenarios of SDARP model are established and solved based on the history or predicted data. The solution results can be used as a reference for many applications such as similar system potential evaluation [Cordeau and Laporte, 2003].
- Dynamic DARP: Contrarily, the Dynamic Dial a Ride Problem (DDARP) with the demand gradually revealed, and requests appear in real time after the operation began. The routes plan need to be constantly adjusted based on the dynamic and stochastic coming requests. Thus, a fast online algorithm is needed to calculate in short time [Cordeau and Laporte, 2003].

Deterministic and Stochastic:

- Deterministic DARP: information including request and traffic information are known before operation [Cordeau and Laporte, 2007].

- Stochastic DARP: uncertain of the information including request and traffic situation before operation [Cordeau and Laporte, 2007].

Number and Homogeneous/Heterogeneous of vehicle:

- Multiple Homogeneous vehicle DARP: a fleet of vehicles will provide mobility services, where all vehicles have same capacity [Cordeau and Laporte, 2003].
- Multiple Heterogeneous vehicle DARP: a fleet of vehicles are used in operation, where vehicles have different types and capacities [Cordeau and Laporte, 2003].
- Single vehicle DARP: only one vehicle is used for operation, usually apply to small scale problem [Cordeau and Laporte, 2003].

Based on the classification, DARPs further extended as single vehicle static deterministic DARP, single vehicle dynamic stochastic DARP, multiple Homogeneous/Heterogeneous vehicle static deterministic DARP, and multiple Homogeneous/Heterogeneous vehicle dynamic stochastic DARP. In this thesis, our study will focus on operating with more vehicle and reservation based demand, which assume that all the traffic information are known in advance, thus it has the same problem setting as the multiple vehicle static deterministic DARP.

2.4.3.3 Multiple vehicle static deterministic DARP

In the course of modelling this problem, most of the previous studies use typical assumption of homogeneous vehicles, one depot, and combine with constraints as time windows of pickup and drop off, ride time, route duration [Parragh et al., 2010, Paquette et al., 2013, Ritzinger et al., 2016]. The objectives of finding the minimum cost of travel time are widely used in this kind of setting. To impose all the constraints, some literature establishes their optimization models with hard constraints where no violation should be considered [Ritzinger et al., 2016, Chassaing et al., 2016]. Nevertheless, others choose the soft constraints rule rather than hard and allow the deviations by adding penalize costs, which show more efficient to find feasible solutions [Cordeau, 2006, Parragh et al., 2010, Kirchler and Wolfler Calvo, 2013, Jain and Van Hentenryck, 2011]. These settings are mainly applied on the algorithm creating and improving.

Additionally, from the modelling perspective, some of the relevant studies consider customers with specific requirements which should served by heterogeneous vehicles [Ilani et al., 2014]. And others take account of the more complex situation which permits transfer activity of passengers from one vehicle to another during their trips [Schönberger, 2017, Blander et al., 2013]. Concerning explore the system maximum potential capacity, Molenbruch et al. [2017] has applied cooperation framework to composites multiple DARP service systems and evaluate the benefits. Likewise, managerial elements are considered by Shen and Quadrifoglio [2013] where centralized and decentralized strategies of DARP para-transit services are applied in the simulation study of Hoston city, it efficiently reduce the total relocation travel time of vehicles and increase the trips can be delivered per hour [Molenbruch et al., 2017]. Simulate the DARP operation and find out that level of service quality setting has serious impacts on the cost of operation.

2.5 Solution methods

Static DARPs are typically considered as combinatorial optimization problem which is NP hardness, and its variables are discrete [Parragh et al., 2010]. The model formulation is based on integer and mixed linear programming with various settings [Stefan Røpke, 2005] and it takes significant computation time to find a feasible solution [Muelas et al., 2013]. For this reason, various algorithms have been studied and proposed on the problem and extension problems which evolved from DARPs. The comparison between the performance of algorithms mainly evaluated upon computation time and quality of results based on same problem setting.

In real life, thousands of requests in the specified period may need to be served by the system. Due to the complexity and scale of the problem, the efficient algorithm needs to be applied and find optimal solutions within a reasonable computation time. Therefore, lots of previous studies focus on this field and try to create the most prorate algorithm to solve the corresponding problem, such as branch and cut [Cordeau, 2006], Tabu search [Cordeau and Laporte, 2003], Genetic algorithm [K. B. Bergvinsdottir, 2004], large neighborhood search [Jain and Van Hentenryck, 2011], etc.

Herein, some of the most important algorithms are listed below according to the different scale of problems.

2.5.1 Exact methods

Exact algorithms for static, deterministic DARP are mainly developed based on the conception of the branch and bound method [Ho et al., 2018a], which can efficiently provide highest quality solutions with small-scale problems in the planning phase. The most important advantage of exact methods is that the solution optimality is guaranteed [Cordeau, 2006].

Various exact methods could be seen as below:

- Branch and cut algorithm (BC): The algorithm is based on the branch and bound procedure with planes cuts added to the branch and bound tree during optimization. The cuts tights Linear Program relaxations and obtain a greater possibility to find integer solutions and provide stronger bounds for verifying optimality [Cordeau, 2006]. This method has firstly proposed by Cordeau and widely used [Parragh et al., 2010, Braekers et al., 2014, Liu et al., 2014] in the further studies.

- Branch and price algorithm (BP): The algorithm is based on the branch and bound procedure with column generation added to the model for Linear Program relaxations. It requires the problem of reformation and split into a master problem and a pricing sub problem. In the model, a set of columns are created to reduce computational dimensions. Due to the reducing feature, it is capable of solving relatively larger mixed-integer formula. Nonetheless, the main drawback is the difficulty of the decomposing original problem to sub problem and ensure the optimality [Veenstra et al., 2017]. This method firstly applied by Garaix et al. [2010], it solved the pricing sub-problem to optimal routes with resource constraints. Feng et al. [2014] applied the method and generated columns in constraint model and efficiently reduce the neighborhood space search for optimal routes.

- Branch-and-price-and-cut algorithms: The algorithm is created by adding plane cuts to branch and price method for the Linear Program relaxations. The algorithm takes advantages of both above-mentioned algorithms by decomposing the problem size with columns generation. It has been used to reduce the paths and solve the sub problem reformed from original [Qu and Bard, 2013, Gschwind and Irnich, 2015].

2.5.2 Heuristics and meta-heuristics methods

Since the exact methods can only be applied to small-scale DARP, heuristic and meta-heuristic methods are developed for larger size problems [Toth and Vigo, 2002]. Heuristics refer to methods which could solve the optimization problem faster and obtain an approximate solution that (in many cases on a theoretical level) approaches the exact solution [Toth and Vigo, 2002]. This kind of methods can be seen as a shortcut which aims to enhance the computation speed by sacrificing part of the solution optimality, completeness, and preciseness in comparison to the exact method [Häme, 2013]. Inside of heuristic method, at each branching phase, there is a heuristic function according to available information, which permutes alternatives in search algorithm and decides the optimal branch to follow Pearl [1984]. Heuristic methods usually perform a limited neighborhood space exploration and obtain good quality solutions within modest computing times [Toth and Vigo, 2002]. Most of the methods could be modified to solve the real-life problems.

Meta-heuristic is (hierarchically) a higher procedure than the basic heuristics methods, it intends to escape the locally optimal solutions during the calculation, compare to heuristics. This is done by enhancing the deeper exploration performance in a larger space of neighborhood solutions. The meta heuristic methods normally constitute of three main functions: a) sophisticated neighborhood space search rules, b) memory structure and c) re-combinations of solutions [Toth and Vigo, 2002]. They can provide higher quality solutions comparison heuristic methods nonetheless requires more computation time.

Heuristics and Meta-heuristics methods can be classified into many types as follow:

- Construction insertion heuristics: simple insertion heuristics is applied to the basic DARP in order to find feasible solutions quickly. Normally it is used for construct the initial solutions in meta-heuristic [Laporte et al., 2014].
- Tabu search: the method is a meta-heuristic method which follows the local search procedure with the additional principle of avoiding the move return to previously neighborhood solutions by applying a Tabu list to memory the history of recent search. Tabu search prevents the situation from falling into a local optimal and perform more extensive exploration of neighborhood solutions. To the best of the author's knowledge, the first use of Tabu search algorithm for the DARP is applied by Cordeau and Laporte [2003] with several diversification strategies. The result of the method application is shown to be very effective and efficient. Afterward, many of the recent studies have used Tabu search algorithm and keep on improving this method on DARPs [Ho et al., 2018b, Paquette et al., 2013, Kergosien et al., 2008, Qi et al., 2008, Kirchler and Wolfler Calvo, 2013, Guemri et al., 2016], which mainly focus on algorithm modification or application with complex real-life DARPs.
- Simulated annealing (SA): is a meta-heuristic method inspired by the physical annealing process, which follows the stochastic local search procedure. At each iteration, one neighborhood solution is evaluated. It also has the function of prevents the situation from falling into a local optimal by accepting non-improving solutions based on the probability. Only a few studies applied this algorithm to the DARPs in recent years [Zidi et al., 2011].

- Large neighborhood search(LNS): is a method that explores a large neighborhood, which enhances the probability of finding local optimal solutions with higher quality in a fast manner. It decomposes the original problem and provides a partial solution, then rebuild it into a complete solution. Each iteration could be seen as the procedure to partially destroy the solution by removing some elements, then repair it by reinserting them. Since this method is relatively effective and fast, many kinds of literature have applied LNS and adaptive LNS methods on DARPs [Qu and Bard, 2013, Dayarian et al., 2016, Jain and Van Hentenryck, 2011, Ghilas et al., 2016].
- Genetic algorithms (GAs): are meta-heuristics methods subsumed on population-based approaches. GAs mimic the biological processes of species and model evolutionary pattern to solve the problem. Algorithms start with an initial population of solutions and conduct four major steps in each iteration as selection, crossover(reproduction), mutation and replacement, then return the best solution. GAs are widely used as clustering method and heuristic search for Dial a Ride routing problems [K. B. Bergvinsdottir, 2004, Masmoudi et al., 2017].
- Hybrid algorithms: are methods which meta-heuristics algorithms integrated with other types of meta-heuristics and mathematical programming. Hybrids of meta-heuristics usually in the form of sequentially run each meta-heuristic algorithm or combine run meta-heuristics algorithm [Laporte et al., 2014]. Many studies have proposed methods on combining meta-heuristics to solve complex problems [Belhaiza, 2017, Masmoudi et al., 2017, Kim, 2011].

2.6 Summary

In this chapter related studies have been reviewed concerning the autonomous vehicle sharing system from different perspectives, such as system definition, components, business model, system design, modeling, operation strategy and solution methodologies. Throughout the vehicle sharing business description and AV definition, it gives a general picture of the concept and components in basic autonomous sharing system. Afterwards,in accordance with the expected impacts on future transportation, the design of the vehicle based autonomous sharing have been clustered into two major types as car sharing and ride sharing, which adopt with various operation strategies. Among them we detailed analyzed reservation based autonomous car sharing and clarify the system component, features, target customers, business model and operation strategy settings, obtain the concept of this system by reviewing previous studies.

Since the optimally routing the shared AVs is significant for successful operation. In order to precisely understand the modelling of vehicle routing process, similar problem setting model of DARP is introduced. In accordance to the past research, we transferred our problem into a general vehicle problem and found the fitting sub-type of DARP. By combining with operation strategy of reservation based demand, multiple vehicle static deterministic DARP shows its perfect match to the reservation based autonomous vehicle routing problem. Moreover, many widely used algorithms are reviewed for problem solution procedure.

Chapter 3

Problem analysis and modeling

This chapter analysis the optimization problem of RACS system and establish the model which effectively reflects the vehicle routing optimization procedure. Four major sections are presented in this part. First section describes the RACS system and clearly defined the optimization problem with the systems setting, operation strategy, objective and tasks. Furthermore, based on system settings, in the next section, an autonomous vehicle sharing (AVS) chain concept has been proposed to represent the autonomous vehicle routing problem, which belong to a combinatorial optimization. Within section 3, features and modelling methodologies of this type of problem has been introduced. Afterwards, a mixed integer programming is applied to mathematical formulation of AVS chains model, which aiming to plan optimal routes according to operation objective and constraints. In the chapter, some research questions have been answered:

- *How to define the autonomous car sharing routing problem?*
- *How to define a "AVS Chain" concept?*
- *What methodologies could be applied to the modelling and what are the decision variables, objective functions and constraints?*

3.1 Optimization problem analysis

The RACS system can be treated as a premier taxi or customized car sharing services. On the former, AV fleet is owned and managed by a taxi company and usually the system capacity (fleet size) would be fixed. On the latter, AV fleet may comprised with private AVs through certain business regulation, which allows the private owned AVs incorporate into the sharing fleet. The fleet size may be constantly varying since some private AVs may not always be available, meanwhile some may join the system when they are idle. Regardless of the ownership, all AVs in same fleet should be managed and operated only by one designated management center and no competition should happen between AVs in same fleet. In such systems, since the reserved requests are known in advance with total awareness of the demand, requests distribution and routes need to be optimally planned and assigned to the AVs before the operation begin. During the planning, AVs will form a group of shared trips on the basis of requests to achieve more significant operation benefits. Particularly, apart from the trip travel, relocation would not be separately considered, which has already count into the routing processes and represented as the sequential of links between the former request drop off location to the next request pickup location. Moreover, there is no need to re-balancing vehicles and heuristically find the next request in relatively high demand areas.

According to the characteristics of the RACS system, assumptions could be made as that an autonomous fleet provides services to a small/large amount of customers. Each customer has made his/her reservation of request with a pair of origin - destination locations, number of passengers, and a preferred departure time. Each customer is supposed to have a tolerance of waiting time for pickup. If the tolerance is exceeded, customers might complain or need compensation by reducing service fee. The operation objective is to plan all the vehicles routes in the road network before the next planning time with the minimum travel time and customers inconvenience.

RACS system optimization problem could be defined with a model which compose of AVS fleet, requests of trips (demand), transport networks, operation strategy and routing process. The detailed system settings considered in this thesis are summarized:

Demand: Given the scope of this master thesis, the complexity of the modelling and traffic data to be considerably simplified, assume that all reservation requests and traffic data in the planning time are know ahead of time [Rigole, 2014].

Network: Road network where system operate to provide mobility services to passengers/customers.

Vehicles: A fleet of homogeneous AVs.

Operation strategies:

1. A fleet of homogeneous shared AVs transport passengers according to their trip demands.

2. In RACS system, customer must reserve the future service based on planning

horizon, such as 24 hours,6 hours, 1 hour or 15 min etc before the trip.

3. Each request may serve one or more people based on the number of accompany passengers, and each AV has its own capacity (seats) limits of loading. In this paper, we assume each request only permit a limited accompany passengers during reservation process, which ensures that the load will not exceed vehicle capacity. Extra passengers can make other requests.

4. Each service of the request include pickup passengers at original node and drop-off at destination node. For each pickup node, a number of passengers within one request are loaded, afterwards, the same number of passengers leave the vehicle at drop off node.

5. Each reservation request should be specified a preferred departure time by customer. We assume that customers all have same limited tolerance of waiting for the vehicle arriving, which denotes as latest departure time. If the AV arriving exceeds the tolerance, customer will get compensation. Departure time and tolerance could be denote as earliest pickup time and latest pickup time which form a time window.

6. Autonomous vehicles are permitted to drive itself without any user inside such as relocation activity;

7. Ride-sharing is not considered in the system.

8. The AVS system has only one depot, all autonomous vehicles are start and end here after all the assigned service provided.

9. In case any AV arrives earlier before the request pickup time, it can always find a temporary parking place and wait for passenger coming.

Objectives: Minimum travel cost and inconveniences cost for customers.

3.2 Autonomous vehicle sharing(AVS) chains

In order to take full advantage of a perfect information circumstances setting with known reservation information, several trips corresponding to requests can be treat as chained to form a route, which could assigne to AV as a period task package. Under this principle, all the reserved trips with spatial and temporal distributions of customer requests could be combined into many linked chains and form the routes according to the optimization objective [Wang et al., 2014]. This concept is called 'trip chain', which is also investigated by Ma et al. [2017] to created optimal routing plan for AVs. For better understanding the

AV routing process, we further split one vehicle trip chain with traversed paths which can be represented as links and states according to the concept. Each of the chain composes of maximum 5 types of links, and also can be expressed as vehicle states in RACS system.

3.2.1 AVS chain definition

5 Links

- START FROM DEPOT LINK: AV is dispatched from the deport to first request.
- WAITING USER COME LINK: Represents the condition of waiting user come, it starts from the vehicle arrival and ends at the time point of pickup.
- TRIP LINK: Represents the trip time length, which is the duration between pickup (start trip) and drop-off the user(arrive at destination).
- RELOCATION LINK: Represents the relocation time length. Vehicle is assigned to one particular request after dispatching process, relocation start from current location and end when arrive at the next request pickup location.
- BACK TO DEPOT: AV back to the deport.

5 States

- START STATE: AV moving with request and no user insider, from depot to request location.
- WAITING FOR USER COME STATE: Stationary of AV with request and no user.
- RELOCATION STATE: AV moving with request and no user inside.
- TRIP STATE: AV moving with request and user inside.
- COLLECTION STATE: AV moving without request and user inside, and back to depot.

The definition of states and links which constitute the trip chains of AVs during the operation period, with the links transition between each other, it efficiently reflects the whole routing process of RACS system operation, which is shown in Figure 3.1.

Based on the link transition process, an autonomous vehicle sharing chain concept (AVS chain) is proposed and expected to apply into the optimization routing planning, the definition of this concept are shown as following:

1. AVS chain: During certain period of time, a series of trip requests have been optimally dispatched and sequentially served by an autonomous shared vehicle

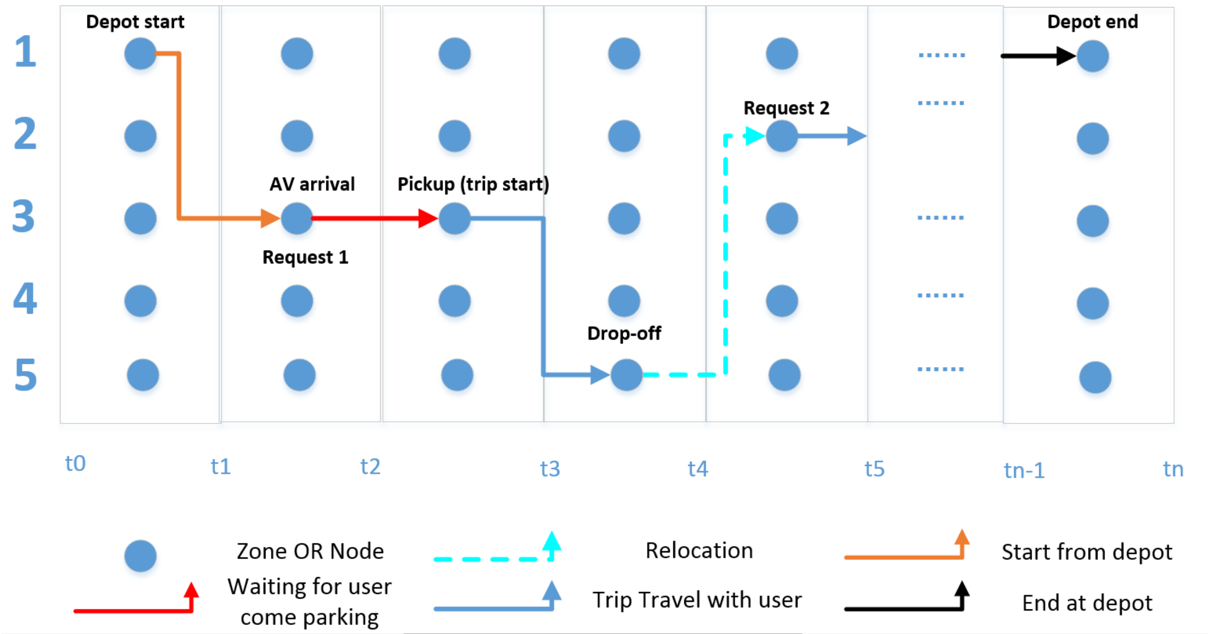

Figure 3.1: Transition process of links

according to the operation objective, the space-temporal trajectory of this AV is a sharing chain. The trajectory consists of links and nodes of requests with spatial sequence connections, simultaneously time sequence between requests occurrence and ending.

2. Multi-AVS chains: During a certain period of time, trip requests have been optimally dispatched and sequentially served by a fleet of autonomous shared vehicles according to the operation objective. The space-temporal trajectories could be represent as multiple AVS chains.

3.2.2 Multi-AVS chains application

As aforementioned, to simplify the problem, we assume that all traffic situation are known in advance. With the reservation strategy, it could be treated the system operating under a perfect information circumstances, where gives the privilege to operators to statically plan the request dispatching and vehicle routing, which could form the optimal AVS chains as fixed vehicle operation routes according to operation objectives. Therefore, in order to maximum the potential of RACS system with this advantage, optimally planning the routes of shared AVs is the most significant task for operation.

It is obviously that AVS chain formation problem is highly related to vehicle routing optimization problem, which has extended features of static, capacited, time constraint, with time window and pickup and delivery. It is similar to a multiple vehicles static dial a ride problem (MSDARP), which denotes as combinatorial optimization. Therefore, we propose and establish the AVS chain model based on the methodology of DARP, where a mixed integer programming could be applied to mathematically formulation procedure. In order to precisely reflect the optimization problem with mathematics modelling, first of all, some of the definitions need to be clarified.

Combinatorial optimization and mixed integer programming

As mentioned above, our problem could be expressed as one special type of multi-vehicle static dial a ride problem. From a modelling perspective, it can be defined as a combinatorial optimization problem.

Combinatorial optimization problem is the model with discrete variables and integer values, and the set of solutions comprised with all possible combinations of value of several discrete variables. Combinatorial optimization problem always consists in identifying the optimal element of a large finite set, where an optimal scheduling and options selection are required [Bierlaire, 2015].

Mixed Integer programming (MIP) is an optimization problem in which objective function and constraints are linear functions of decision variables, and some of the variables are restricted to take integer values and some are allowed to take non integer values. Integer variables make optimization non-convex as NP hard problem and difficult to solve. Memory and solution time rise exponentially with problem size. A mixed-integer linear programming problem (MIP) usually with the form as

$$min_x cx \tag{3.1}$$

subject to

$$Ax \leq b \tag{3.2}$$

where

$$x \in \mathcal{Z}^n * \mathcal{R}^p \tag{3.3}$$

The set S of all $x \in \mathcal{Z}^n * \mathcal{R}^p$ which satisfy the linear constraints $Ax \leq b$ S $= \{x \in \mathcal{Z}^n * \mathcal{R}^p$, $Ax \leq b\}$ is called feasible set. An element $x \in S$ is feasible solution. Objective function and all constraints are linear, some variables are integers, some variables are continuous [Bierlaire, 2015].

3.3 Formulation

Based on the principle of mixed integer programming and combinatorial optimization, our AVS chain model could be formulated as follow:

Notation of parameters and variables:

N = total number of requests;
$P = \{ 1, ..., n \}$, set of pickup location nodes of requests;
$D = \{ n+1, ..., 2n \}$, set of drop off location nodes of requests;
$G(V, A)$ - Directed network graph, with V set of nodes and A set of arcs;
$V = \{ P, D, \{0, 2n+1\} \}$, set of all nodes in the network. $0, 2n+1$ are the start depot node and terminal depot of the system;
A = arc set, $\{$(i,j) $: i = 0, j \in P, or\ i, j \in P \cup D, i \neq j\ and\ i \neq n+j, or\ i \in D, j = 2n+1 | \forall i, j \in V\}$;
e_i- earliest pickup time of r_i at node i, which is the preferred departure time of customer;
l_i - latest pickup time of r_i at node i, where is the acceptable tolerance of customers for waiting;
r_i- user request, request r_i consist of pickup node i and drop off node $n+i$, the pair of nodes $(i, n+i)$ denoted to one request, $i \in P, n+i \in D$; k- vehicle k, $k \geq 1$, $k | \forall k \in K$;
K- fleet size of the AVS, $K \in \mathcal{N}$;
TW_i^k - costumer waiting time for delay arrival of vehicle k at node i, could be see as inconvenience cost for customer;
α - coefficient of waiting time(punishment coefficient);
c_{ij} - travel time or cost of traverse on the arc of (i, j);
T_{max}^{du} : maximum route duration for all vehicles;
B_i^k - When vehicle k starts visiting node i (arrival time);
T^p- planning period time of AVS system;
L_i^k - The ride time of request i on vehicle k;
L - The maximum ride time of any request;
x_{ij}^k - binary variable which indicates the vehicle k traverse the arc (i, j) or not, if traverse value 1, otherwise 0.

The model is defined based on demand characteristics, travel time (trip time and relocation time), routing plan, number of depots, time windows.

AVS chain Model additional assumption:
Assumption 1. *Generalized transport costs include: AVs trip time and relocation time;*

waiting cost, which represents the penalties for arriving early or late to each trip destination.

Assumption 2. *Planning period time: System only provide single planning period time, where the system is aiming to fulfill requests within a specific interval of time period. Consequently, demand and required traffic conditions are assumed to be available ahead of planning.*

Assumption 3. *Travel times c_{ij} are subject to the time period of the day. In static planning, travel time matrix between each link are pre-calculated based on historical or prediction data in the network and we suppose that both direction travel time are same. For each period of time, given link travel time we can calculate the expected travel time between OD links and relocation links of requests.*

3.3.1 Inputs

In order to allow for a proper representation of the system and to apply the required optimization process, relevant inputs need to be defined, which include network for the examination of travel times and trips, demand, vehicles and depot.

3.3.1.1 Transportation network

An essential element of the model is a traffic network of the study area, where time is defined for a planning period of T^p. Aiming at expanding a spatial network to a temporal-spatial network, we modify it in terms of links combination as one path which directly represents the path from one request pick up node to drop off node. To simplify the model, we consider a network with real paths represented by a directed graph $G = (V, A)$, where $V = \{v_1, v_2, ...v_{2*n}, v_0, v_{2*n+1}\}$ represents the set of all nodes in the network. $\{v_1, v_2, ...v_n \}$ is further specified to represent the requests pickup locations P and $\{ v_{n+1}, v_{n+2}, ...v_{2*n}\}$ as D that represents the requests drop off locations. $\{v_0, v_{2*n+1}\}$ represent the depot. It should be noted that we simplify the model specification by utilizing only one depot. Essentially, v_0 and v_{2*n+1} refer to the same location in the network. Moreover, A is the set of arcs with travel time between all nodes. The arc from $i \in V$ to $j \in V$ is denoted by$(i, j) \in A$.

3.3.1.2 Demand

In the planning period T^p, let N denote the number of requests from the customers. Each of the request has an i and it is split into the pairwise representation, with a pickup and drop off request $(i, N+i)$, which correspond to a pickup and drop off node (v_i, v_{n+i}) (Section 3.3.1.1). The earliest and latest pickup time of request i (time window) is attributed to the pick-up node v_i as (e_i,l_i). The pick up location and drop off location could also refer to zones or stations, which represent the origin and destination locations of similar requests. There are no customer priorities, neither passengers transferred between vehicles.

3.3.1.3 Vehicles and depot

A homogeneous fleet of K AVs are provided, with k referring to the random vehicle. Location of depot as well as number of available vehicles are known. Violation of vehicle capacity is safeguarded by the properties of the reservation system, which limited the number of passengers for each request.

3.3.2 Variables

Decision variables
Based on the above definitions and assumptions, following decision variables are defined.

x_{ij}^k - Binary variable:

$$x_{ij}^k = \begin{cases} 0 & Vehicle\ k\ not\ traverse\ through\ the\ arc\ of\ (i,j) \\ 1 & Vehicle\ k\ traverse\ through\ the\ arc\ of\ (i,j) \end{cases} \tag{3.4}$$

Fractional variables
B_i^k - When vehicle k starts visiting node i(arrival time).
TW_i^k - Waiting time of each request

3.3.3 Objective function

Finding a set of AVS chains which are represent as routes of vehicles start and end at the depot with minimize the total travel cost including the trip link, relocation link and violation value as customer waiting time TW_i^k.

$$Min\ (\sum_{k\in K}\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij}^k + \sum_{k\in K}\sum_{i\in V} TW_i^k) \tag{3.5}$$

3.3.4 Constraints

Precedence Constraints

1. Every request is served exactly once:

$$\sum_{k\in K}\sum_{j\in V} x_{ij}^k = 1, \forall i \in P \tag{3.6}$$

2. Same vehicle serves a pair of pickup and drop-off nodes:

$$\sum_{j\in V} x_{ij}^k - \sum_{j\in V} x_{n+i,j}^k = 0, \forall i \in P, k \in K \tag{3.7}$$

3. Every vehicle leaves the source node:

$$\sum_{j\in V} x_{0j}^k = 1, \forall k \in K \tag{3.8}$$

4. Same vehicle enters a node and leaves the node:

$$\sum_{j\in V} x_{ji}^k - \sum_{j\in V} x_{ij}^k = 0, \forall i \in P \cup D, k \in K \tag{3.9}$$

5. When the end of each chain/route, the vehicle go back to depot:

$$\sum_{j\in V} x_{i,2n+1}^k = 1, \forall k \in K \tag{3.10}$$

Requirement / Order Specific Constraints

1. Setting and checking the visit time for Time Window:

$$B_j^k \geq (B_i^k + c_{ij})x_{ij}^k\ , \forall i \in V, j \in V, k \in K \tag{3.11}$$

$$e_i \leq B_i^k \leq l_i, \forall i \in V, k \in K \tag{3.12}$$

2. Set and check the ride time:

$$L_i^k = B_{n+i}^k - B_i^k \ , \forall i \in P, k \in K \tag{3.13}$$

$$L_i^k \leq L \ , \forall i \in V, k \in K \tag{3.14}$$

   Since we do not have a mandatory requirement to the ride time of each request, here we could set the L equals to T^p

3. Binary variables:

$$x_{ij}^k \in \{0, 1\} \ \forall i \in V, j \in V, k \in K \tag{3.15}$$

4. Waiting time of customer

$$TW_i^k = B_i^k - l_i \tag{3.16}$$

Equation 3.5 presents the objective function of minimum total travel time cost of all routes and customer waiting time. According to the precedence constraints, Equation 3.6, 3.7 ensures that for each request i, both the pick-up and the drop-off nodes (v_i, v_{n+i}) belongs to same route and only can be served once with the same autonomous vehicle k . Meanwhile, constraints 3.8 and 3.10 indicates that each autonomous vehicle starts and ends its route/operation at the depot. The constraint 3.9 denotes that each of the vehicle which enters a node leaves it as spatial sequence. Moreover, the requirement constraints 3.11 and 3.12 guarantee that for every request i, pickup node v_i is visited before drop off node v_{n+i}, and each service of the request i at node v_i needs to start its pickup activity in the interval of time window $[e_i, l_i]$. Constraints 3.13 and 3.14 indicate that each ride/trip time of the request is the difference between the pickup time and drop off time, and it should not exceed the maximum ride duration.

Moreover, constraints 3.11 and 3.12 determine that the time window would be violated if $B_i^k \geq l_i$, and the violation value denotes as waiting time of customers. If the vehicle arrive at the request pickup node v_i earlier than the earliest pickup time e_i, autonomous vehicle need to temporarily parking and wait for the customer coming. Equation 3.16 is the waiting time of customer for delayed pickup which violate the time window.

It should be noted that in this model, we consider a soft time window constraint which allow the violation, it permits the waiting time of customer for vehicle to come after the latest pickup time. When the waiting time of a customer is increasing and can not be eliminated during the optimization process, this situation could be seem as system capacity insufficient, where the current number of vehicles could not fulfill all the requests, and operator should consider increase the fleet size. Nevertheless, in reality, companies will not constantly increase their system capacity based on increasing of demand, so we assume that RACS system applies a potential service level standards which indicates that the average request waiting time should be limited within certain range as Level of service (LOS), such as 5-10 min. If the violation exceeding the limit, operator must add vehicles to enhance the service quality. Therefore, a average waiting time is proposed, which could be represent as

$$ATW = (\sum_{k \in K} \sum_{i \in V} TW_i^k / N) \leq LOS, \forall i \in V, k \in K \tag{3.17}$$

where ATW is average waiting time, N is the total number of requests, and LOS represents the level of service, which unit is minute.

This evaluation will be applied at the end of optimization process where the output waiting time results will be divided by total number of requests. It will be assessed with LOS value. If exceeding, vehicle added, the optimization process and calculation repeats until fulfill the standard.

Chapter 4

Solution methodology

Our problems are combinatorial optimization and do not allow high dimensions calculation since the computation cost will increasing exponentially with the dimension expanding, and feasible solutions may not be able to obtained. Therefore, decomposing the initial problem size into multiple smaller sub problems is an option for solution, so the clustering and meta-heuristics methods could be combined and applied into the solution procedure.
In this chapter, we explore solution methods by proposing meta-heuristic Tabu search in combination with clustering methods of K-Means and K-Medoids algorithms into the RACS routing optimization model solution procedure. First of all, we explained the meta-heuristic method advantages and applicability with the combinatorial problem as our model. Then detailed discuss the Tabu search function and application with its features including neighborhood structure, initial solution construct, Tabu search iteration, neighborhood evaluation and reduction, and insertion technical. Afterwards, following the K-Means and K-Medoids methods application steps in the solution process. In the chapter following research questions is addressed
How to solve the model with appropriate algorithms with different scale of demand problems?

4.1 Meta-heuristic method

Heuristic and Meta-heuristic methods have been widely used in the NP-hard combinatorial problems similar to RACS optimization problem. In contrast to the exist exact algorithms which only can solve small scale problems, meta-heuristic algorithm could obtain solutions for increasingly large and complex problems. Meta-heuristic method is a higher level procedure than the heuristic method, which aims to escape the local optimal solutions that in many cases occurs during the calculation. As with heuristics, meta-heuristic has the advantages of decreased computation time than exact methods, nevertheless, the drawback is that there will be no guarantee to obtain the optimal solution. In this thesis, in order to find feasible solutions for real life problem within reasonable computation time, we applied meta-heuristic method of Tabu search combined with K-Means and K-Medoids clustering algorithms into the different size of real life problems.

4.1.1 Tabu search

Tabu search use local search or neighborhood search iteratively proceeding from one feasible solution to another improved one until the max iteration reached. Each solution has various neighborhood solutions, the search procedure of Tabu is called move, which represents from potential solution x to its neighborhood solution x'. Tabu search allow the moves deteriorate current objective function value in order to escape the local optimal and avoid the pitfalls of local search stuck in poor-scoring areas. During the search progress, larger space and new neighborhood will be explored, where memory structure as tabu list will be applied to guide purposes and determine the new neighborhood solutions. Usually, memory structure is a set of rules used for filter the feasibility solutions during the exploration, which works as record the attribute move from one solution to another [Glover, 1989].

Principle of Tabu search:

1. Begin with random solutions.
2. During iterations, the move construct and destruct solution where new solution will be generated.
3. Perform intensification which is based on objective function and exploration phases.
4. Perform diversification, objective function value can be deteriorate.
5. Modify solutions in new area which has been explored.

As aforementioned in section 2.5.2, Tabu search is a meta-heuristic algorithm which can be applied into combinatorial optimization problems solution procedure, such as traveling sales man problem [Toth and Vigo, 2002] as well as the pickup and drop off problem with time window.

Tabu search intends to progressively explore the neighborhood framework from the initial solution. In order to escape the local optimal, during the procedure some similar attributes of recently visited solutions are declared forbidden, or put in tabu list for some iterations which is regulated by the operator. During processing, a continuous diversification procedure is running to reduce the probability of falling into local optimal. It can be treated as a guided local search method which is efficient in exploring the search space to find a global optimal solution for combinatorial optimization problem [Ho et al., 2018b].

4.1.2 Tabu search in RACS routing problem

The use of Tabu search in the RACS problem requires the specification of some mechanisms and structures. The subsections bellow provide a detailed description of the the various components that are required to be defined in order to make Tabu search for RACS applicable, which is based on the methodology of Cordeau and Laporte [2007].

4.1.2.1 Relaxation mechanism

Before the Tabu search, a relaxation proecss will be proposed for the objective function which allow the violation of the time window constraints and obtain the probability to explore infeasible solutions in the search process. The solution process goes as follows: first generate the solutions S in which each $s \in S$ is a set of k routes, it is intent to satisfy the constraints of 3.6, 3.7, 3.8 3.9 and 3.10 that represent each route start and end the service at depot, each request only served once with same vehicle, pairwise pickup and drop off nodes are visited sequentially. The solution of s may have violation of the time window constraints associated with the requests.

In the objective function $\sum_{k\in K}\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij}^k$ denote to be the total travel time/cost of all vehicle routes, and $\sum_{k\in K}\sum_{i\in V} TW_i^k$ is the total waiting time for customers which denotes violation of the time window constraints at each node. Solutions are evaluated by applying the relaxed objective function

$$f(x) = \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij}^k + \alpha * \sum_{k\in K}\sum_{i\in V} TW_i^k \tag{4.1}$$

where adding α to multiply with violation time, which is a self adjustment parameter behaving as a punishment coefficient. During the search process, the value of α will be dynamically adjusted, following a proportional increase to the violation magnitude. This penalty allow the enlargment of the solution space exploration and provides a simple modifications through intermediate infeasible solutions [Cordeau and Laporte, 2007].

4.1.2.2 Neighborhood structure

In the search process, we defined attributes as $U(s) = (i, k)$, i represent the request and k represents the vehicle ID or route ID, which is aiming to characterize the solutions $s \in S$. Neighborhood solutions $N(s)$ of s is composite with all solutions which can be obtained from s by applying simple modification. The procedure is removing an attribute (i, k) and replacing it with another one (i, k'), k is not equal to k', it means that remove the request i from route k to route k'. Correspondingly, the pairwise pickup and drop off nodes (v_i, v_{n+i}) are deleted from k and they are moved to another route k'. By inserting the (v_i, v_{n+i}), route k' then is performed to minimize the total value increase in objective function by simple insertions.

Attributes are applied to control the Tabu search and deliver a diversification strategy in the Tabu search process. After the removal, this activity of attribute (i, k) will be in the Tabu list and the reinsertion in that route will be forbidden for next few iterations. When the tabu iterations has passed or when it could lead the search to get a smaller value of the objective function than the previous best, the attribute will be removed from the list and revoked.

4.1.2.3 Initial solution construction

The first step of the Tabu search is to create a initial solution s_0, which is composed with each request i assigned to a randomly vehicle. This process including the insertation of each pair of pick up and drop off nodes, sequentially into each route corresponding to a vehicle. During the insertion, constraints of 3.6, 3.7, 3.8, 3.9 and 3.10 are guaranteed, nevertheless, other constraints may have violations.

4.1.2.4 Tabu search iteration

During iteration calculation, the algorithm will start from the initial solution s_0 and proceed selecting the best solution, which is not in Tabu list, according to the relaxed objective function which is presented in section 4.1.2.1. At each iteration t, α value will be adjusted by a factor of $1 + \sigma$. If the current solution is feasible with the time window constraint, the value of α will be divided by $1 + \sigma$, $\sigma \in [0, 1]$. In contrast, if it is a non-feasible solution, value of α will be multiplied by $1 + \sigma$, $\sigma \in [0, 1]$. The process will be repeatedly run and the best solution that has been found during its search will be denoted as the final solution.

4.1.2.5 Evaluation of neighborhood

The purpose of neighborhood evaluation is to reduce the violation of constraints and assess the feasibility of solution. Based on the insertion technical, the attribute of (i, k) removed from the route k and inserted to route k' will have certain impact to the feasibility of time window for all requests in those two routes. A substantial recalculation process needs to be executed to evaluate the impact of value changes to the objective function. For instance, within a complete route k= $\{v_0, ..., v_i, ...v_t\}$, v_0, v_t denotes the depot, in order to calculate the time window violation, the arrival time of A_i and departure time of B_i at request node i are necessary. Thus, the value of both indicators will be changed and need to be updated to recompute the violation value. The impact of removing the request from route k and insert to specified position of route k' can be evaluated by following processes to each routes involved in the insertion and remove exchanges.

1. Set $D_0 = e_0$, D_0 is the departure time from depot.
2. Calculate the Arrival time A_i, service start time of B_i, for each request node i in the route.
3. Calculate the violation value of time window as waiting time.
4. Update the changes in violation of constraints.

It is necessary to notice that with the process execution, although the procedure yields to optimally arrange the departure and arrival time for the objective of minimizing the time window violation, it is still require to consider the value of the coefficient α to find an optimal solution for evaluation (see in section 4.1.2.1).

4.1.2.6 Reduction of neighborhood

For each pair of pickup and drop off nodes (v_i, v_{n+i}) for request i, if the pickup time satisfies the condition of $e_i \neq 0, or\ l_i \neq T^p$, where node of v_i denotes a critical node (section 4.1.2.7) which has a time window constraint. In order to reduce the size of neighborhood solutions, we consider each iteration of Tabu search follow the rule of evaluation insertion process.

1. First determine the best insertion position for critical node

2. Then with the pairwise principle, determine the other corresponding node in its best position.

This will dramatically reduce the exchanges for request i form $O(m^2)$ to $O(m)$, in which m is the number of node in route k. Each of the exchanges only involves two routes, other routes cost are not changed during the insertion and deletion cost process.

4.1.2.7 Insertion technique

The computation time of the problem is strongly effected by the size of neighborhood solutions which are determined by the insertion technical. In the thesis, two step insertion and reduction of neighborhood technique is adopted in the optimization process due to the nature of solution transition, which could be seen in Figure 4.1. The algorithm is moving the pairwise pickup and drop off nodes to another route. In the request, we denote the pickup node as the critical node with a tight time window. Firstly, insert the critical (pickup) node in the best position and get a objective function of least value, then hold the position and insert the non critical node (drop off) to its corresponding position. This technical could reduce the neighborhood size from $O(m^2)$ to $O(m)$ and shorter the computation time.

4.2 Clustering method

Tabu search algorithm is considered to be efficient in finding the solutions for relatively small scale problem [Toth and Vigo, 2002], nevertheless, since the model is an NP hard, when it applies to large scale problems, the computation time could be dramatically raised with the calculation dimensions. Clustering is a method to decomposing the original problem into several subsets. With the large size problem, clustering method has the advantages of reducing the calculation time.

Operating companies of RACS system may need to plan the routes with large amount of requests in a short period of time, raising the necessity to speedup the calculation process. Therefore, with the application of clustering methods the decomposition of the large size into many smaller subproblems and implement the parallel calculation to reduce computation time become possible. It is an alternative to enhance the operation efficiency.

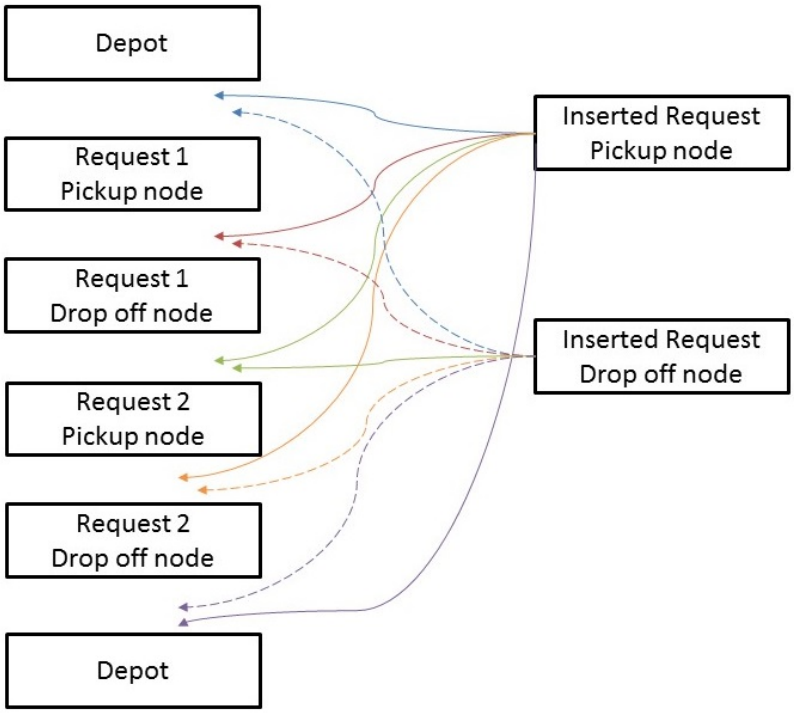

Figure 4.1: Insertion technique

4.2.1 K-Means clustering

K-Means method is a partitional clustering approach for decomposing the problem into independent subsets, which follows the principle of finding clusters of data based on their similarity. It is an unsupervised learning algorithm which have no predictable outcome and only intents to find patterns within the dataset. In the clustering procedure, preferred number of clusters should be specify in the algorithm before execute [Lucasius et al., 1993]. Each observation (data points) will randomly assign to a cluster, and find a set of centroids for each group of observations. Afterwards, iterates will be done for optimization the clusters with following steps:

1. Reassign data points to the cluster whose centroid is closest.
2. Compute and select new centroid for each cluster.
3. The variation of cluster is calculated with euclidean distance or other metric methods between each data points and their corresponding centroids.
4. Repeated the first two steps until the variation in cluster are not decreasing any more.

Based on the algorithm principle, K-Means is applied in our model to decompose the request demands.

We have clustered the demand according to each request initial departure time and form a pairwise feature as a request node with earliest pickup time and drop off time $[e_i, e_i{+}c_{i,i+n}]$, where we assigned $Pt_i = e_i$, $Dt_i = e_i + c_{i,i+n}$ ($c_{i,i+n}$ is the travel time). Therefore, each request could be represented as a node $R_i(Pt_i, Dt_i)$. In dataset, it is a n*2 matrix of all request nodes. In order to define the clustering methods, let $X{=}x_1, x_2, ...x_n$ be the total observations with n elements, k is the number of clusters, λ is the set of centroids and $S{=}S_1, S_2, ...S_k$ is the total observations n partition into k groups.

With the definition of the request matrix, K-means algorithm could be implemented by importing the dataset and setting the number of clusters k based on the subproblem size need. During the calculation process, we applied the city-block metric, and the formulation of objective function is $D(x, \lambda) = argmin_X \sum_{x \in S_i} |x_i - \lambda_i|$. The function represents the minimum absolute differences sum between points and its cluster centroid, where each centroid is the component's median of the points in that cluster. The goal is to minimize sum in cluster with the nearest mean.

4.2.2 K-Medoids clustering

Similar to K-Means, the K-Medoids algorithm is also a clustering method, created based on K-Means and the medoid shift algorithm. It is a partitioning approach which splits the initial dataset into k clusters and aims to minimize the distance between the data points and the point designated as the center of same cluster group, where points are predefined in the problem.

K-Medoids algorithm selects the centers/medoids based on data points and defines the distances between the points by applying a distance metric, such as Manhattan Norm, Euclidean, Seuclidean, Cityblock etc. It should be mentioned that the cluster group setting k are known in prior to the algorithm running.

Compare to K-Means method, the K-Medoids is more robust to the noise and outliers, since it could minimize a pairwise dissimilarities distance rather than squared distances [Muelas et al., 2015].

4.2.2.1 K-Medoids formulation

In oder to decompose the initial large size of requests data, K-Medoids clustering method is proposed to be incorporated in the discussed optimization model (Section 3.3) to create k groups of the requests. Each group contains similar sets of elements. The principle of this method is to ensure that closer elements are assigned to the same group and improve the final results, where the elements are requests. In short, given a set of observations X=$x_1, x_2, ...x_n$ with n elements, K-medoids method will split the X into k sets ($k \leq n$). $S = S_1, S_2, ...S_k$ by applying the partitioning around neighbors algorithm (PAM) under K-medoids framework with defined distance measure. Let X=$x_1, x_2, ...x_n$ be the total observations with n elements, β be the set of the medoids, I_β be the set of indices of the medoids elements in X, $I_\beta \in X$. $I_{X-\beta}$ be the indices of elements which are non-medoids in X, $I_{X-\beta} \in X$. I_X be the set of indices of all elements, where $I_X = I_\beta + I_{X-\beta}$. Also let u_{ij} be the binary variable which indicates the distance is minimum or not. $d(x_i, x_j)$ is the distance between element x_i and x_j. β_{ij} is the neighborhoods of β, where $x_j,\ j \in I_{X-\beta}$ replace $x_i, i \in I_X$.

If exists two set of medoids β and β', both of them consist k elements, and share with $k-1$ elements, β and β' can be seen as neighbors. Set of medoids β with k elements could have number of $k*(n-k)$ neighbors. and β_{ij} represents the neighbor of β if $x_j,\ j \in I_{X-\beta}$ replace $x_i, i \in I_X$. The difference between neighbor solutions can be formulated as $\nabla F_{ij} = F(\beta_{ij}, U_{ij}) - F(\beta, U)$

Variable: u_{ij} - Binary variable :

$$u_{ij} = \begin{cases} 1 & if\ the\ distance\ of\ d(x_i, x_j) = min_{q \in I_\beta} d(x_i, x_q),\ i = 1, 2, ...n \\ 0 & otherwise \end{cases} \tag{4.2}$$

The objective of the model is aiming to provide a quality clustering groups with a given set of medoids β, which could best represents the data set of X, the objective $F(\beta, U)$ should be minimum.

$$F(\beta, U) = \sum_{i \in I_{X-\beta}} \sum_{j \in I_\beta} u_{ij} d(x_i, x_j) \tag{4.3}$$

4.2.2.2 Partitioning around medoids algorithm

A solution method for the K-Medoids clusterring is the Partitioning Around Medoids (PAM) algorithm. It usually utilizes heuristic greedy search to find the optimal solution with fast calculation speed. The PAM algorithm process is presented as bellow:

1. Building, selection of initial medoids: from the initial dataset n, select k points as the medoids.
2. Assign every non-medoids data point to the closest medoid.
3. Calculate the objective function of total sum of distances, and evaluate the total distances decrease or not.
4. If decreasing:
   - for each medoid and each non-medoid data point, swapping medoid and non-medoid point, and recalculate the sum distances of points to their medoid.
5. if increasing in previous step:
   - stop the swapping.

PAM algorithm consists of two steps as a building and a swapping procedure, which are based on different incremental stepping heuristics. During the PAM algorithm, it start with a set β with randomly selected k medoids, among the neighbors $k * (n - k)$ to find optimization solutions, the whole procedure can be present as follow:

1. For all neighbors $\beta_{ij}, i \in I_\beta, j \in I_{X-\beta}$ of set the β, select $\beta_{qr}, q \in I_\beta, r \in I_{X-\beta}$.
2. Calculate the $\nabla F_{qr} = F(\beta_{qr}, U_{qr}) - F(\beta, U)$ and get the result of ∇F_{qr} with $\nabla F_{qr} = min_{ij} \nabla F_{ij}$.
3. If $\nabla F_{qr} \geq 0$, the calculation has reached a local optimal minimum and stops. and if $\nabla F_{qr} < 0$, the set of β will be replaced by β_{qr} then the same calculation process repeats.
4. When the algorithm reached max iterations, each non-medoids element $x \in X - \beta$ will be assigned to clustering groups which are represented by the medoids.

4.2.2.3 Distance metric method

Since the PAM calculation is based on the distance metric between the elements, its selection is an important process for successfully deliver this model. In our problem, distance between elements is seen as the pairwise dissimilarity, which is a distance matrix created for each unique pair of requests. Due to the computation time, it is efficient to perform the calculation in parallel. The matrix of request distance could be directly applied to the K-Medoids algorithm in the central node (can be the first node) to obtain the clustering groups. Then send to other computing nodes. Afterwards, each of the computation node will construct its own m/k routes solutions, m represent vehicles and k represents the number of clustering groups (computation nodes).

The distance between the requests needs to be appropriately measured since each request is structured according to the pair of pickup and drop off locations. So as to measure the distance between them, a proper metric should consider both locations. In this thesis we defined our metric based on method of presented by Muelas et al. [2015], which consider of four possible positions of requests and calculate the shortest route for traversing all requests. This method fulfill the requirement of metric conditions such as triangular inequality [Muelas et al., 2015]. The position relationship between two requests $R_i(i, i+n)$ and $R_j(j, j+n)$ are describe in figure 4.2.

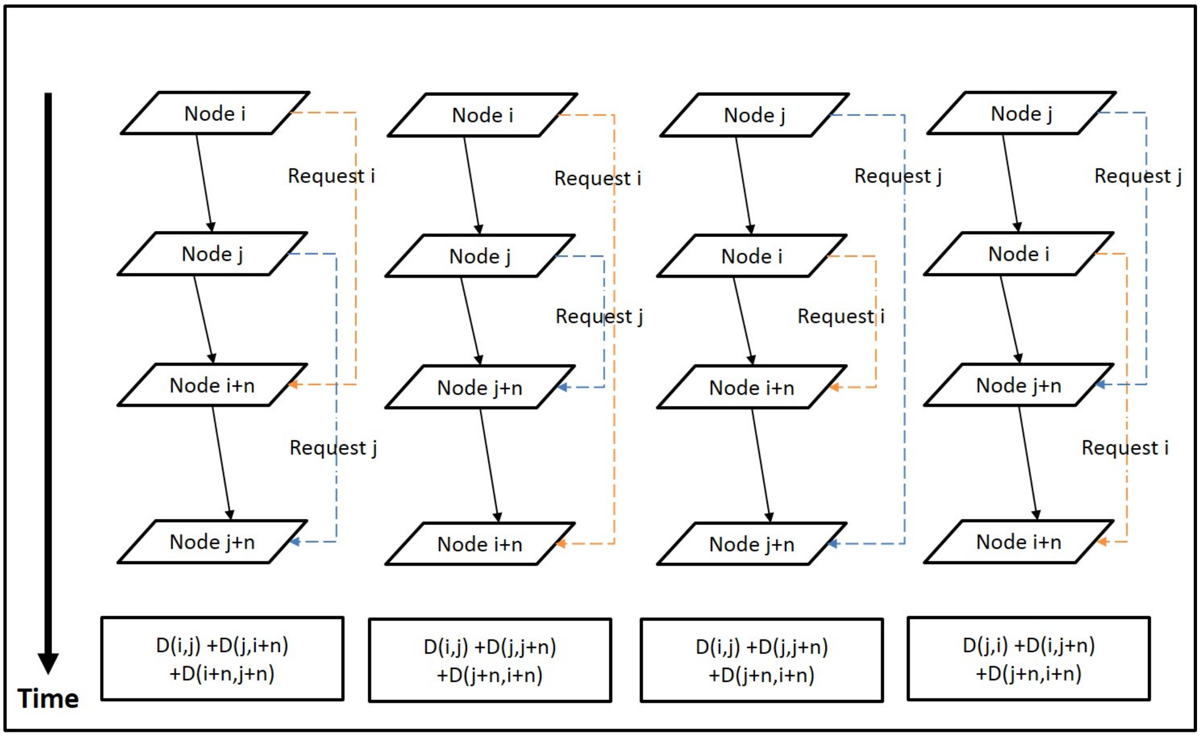

Figure 4.2: Request position relationship

Simultaneously, the minimum distance formulation between pair of requests could be defined as equation 4.4. Let R_i describe request ID i with pair of pick up node i, and drop off node $i+n$. R_j is the request ID j with pair of pick up node j and drop off node $j+n$, and $D(R_i, R_j)$ is distance between two requests of R_i, R_j.

$$\begin{aligned} D(R_i, R_j) = min(&D(i,j) + D(j, i+n) + D(i+n, j+n), \\ &D(i,j) + D(j, j+n) + D(j+n, i+n), \\ &D(j,i) + D(i, i+n) + D(i+n, j+n), \\ &D(j,i) + D(i, j+n) + D(j+n, i+n)) \end{aligned} \tag{4.4}$$

Chapter 5

Solution algorithms

This chapter presents the processes of implementing the three algorithms into a model solution procedure, and explains the function of each algorithm step by step. In this chapter the following research questions are addressed
How are the algorithms applied in the solution process?

5.1 Tabu algorithm

For the solution procedure three algorithms are applied including Tabu, K-Means Tabu and K-Medoids Tabu methods. The flowchart of algorithm application could be seen in Figure 5.1. The next subsections present practical considerations when developing the components for the application.

5.1.1 Objectives and constraints

The RACS model consists of m vehicle routes with corresponding requests on network G. By assigning shared autonomous vehicle trips to the network, the output of a set of AVS chains is obtained as well as requests sequences with minimum total cost of travel time and customer waiting time. The optimization process needs to fulfill the following constraints:

1. Every route starts and ends at the depot.
2. For every request i, node v_i and $v_i + n$ belong to the same route and node $v_i + n$ is visited after v_i.

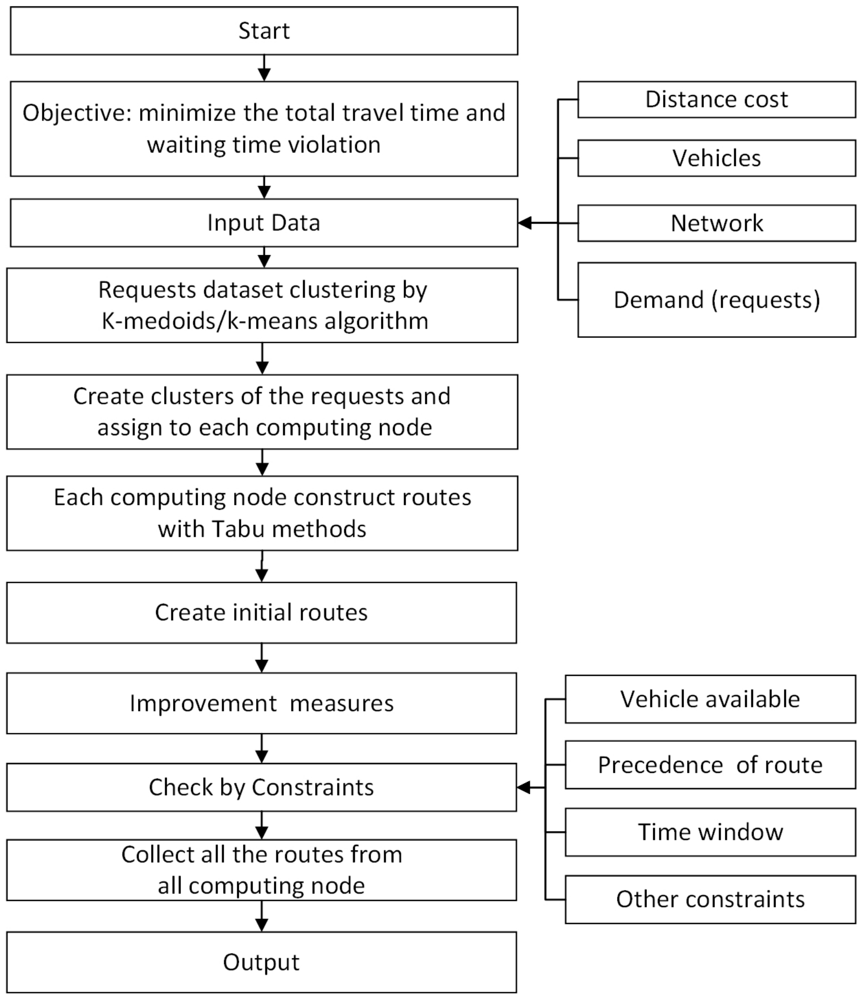

Figure 5.1: Flowchart of algorithm

3. The service at vertex v_i begins in the interval $[e_i, l_i]$, and every vehicle leaves the depot and returns to the depot.

4. The ride time of any user does not exceed L,in our problem we set $L = T^p$.

5.1.2 Input dataset structure

A dataset, which consists of network coordinates, requests and distances, needs to be imported in the derived procedure.

Structure of Request data

Original requests dataset is a n*4 matrix, which contains pickup node, drop off node, earliest pickup and latest pickup time.

(pickup node— drop off node — earliest pickup time —latest pickup time)

In order to simplify data structure for programming, herein the original request matrix is modified and convert to a $2n$*3 matrix, which split the pairwise pickup and drop off nodes of one request into two requests where each request contains one node. The structure of new matrix formed as ID 1 to n, which represent all pickup nodes. Correspondingly, ID $n+1$ to $2n$ represent all the drop off nodes. Request i and request $n+i$ belongs to one pair.

Structure of cost matrix

Cost matrix contains the distance matrix between pairwise nodes, here we define the distance as travel time. Based on the assumption that the system operates under perfect information circumstances, all travel times are known in advance.

Structure of coordinates:

The coordinates dataset contains pairwise coordinates represents the nodes in network $G(V, A)$. (X-coordinate —Y-coordinate)

5.1.3 Algorithm application

Main algorithm:
STEP 1. Input Network $G(V, A)$, matrix *Requests*, Vehicle k, Cost matrix *Distances* .
STEP 2. Setting parameters α, σ, $tabuMax$, max *iterations* and max duration of routes-planning horizon.
STEP 3. Construct Initialization Solution S_0 with randomly assign all the requests to vehicles.

- Set the depot location and start all routes from the depot.
- Sequentially assign the pick up node, earliest and latest pickup time (e_r,l_r) of each request r to random vehicle.
- All routes end at depot.
- Generate initial route *Routes* and request matrix $requestMat$.

STEP 4. Find each request pickup time and customer waiting time.

- Use $findTime$ function, calculate each request pickup time, vehicle idle time based on *Routes* and $requestMat$;
- Calculate the travel cost of all routes;

- Calculate the total customer waiting time $newWaits$ by applying $waitCalculator$ function.

STEP 5. Save all initial value and routes as $holdRoutes, initialRoutes, Objective, pickuptime$
STEP 6. Tabu search iteration. When the iteration is not reach maximum, repeat all procedure from step 7 to step 17.
STEP 7. Search each car route with pairwise setting since each customer has origin and destination locations;
STEP 8. Establish the tabu check with the tabu list when algorithm pass 300 iterations, check each vehicle in tabu list, if the request are in the same spot of the list, tabu become ture, end the section. Otherwise false, assign i as location prior, $j1$ as candidate pickup route location, $h1$ as candidate pickup request ID, $j2$ as candidate drop off route location, $h2$ as candidate pickup request ID, and k location after the current request.
STEP 9. For each other car (don't search the same car) and save the candidate pair. Check the previous destination and the current origin spot where can insert a new route pair;
STEP 10. Calculate the assumption insertion move, the distance savings difference as $newSavings$ and $tmpSavings$. Create dummy routes to compare the potential moves;
STEP 11. Calculate the costs of the move by measuring the cost of the original against the potential move by applying the $Moves$,$findTime$ and $waitCalculator$ functions, find $pickupTemp, wait, waitDiff$ and $newWaits$;
STEP 12. Evaluate the cost $bestSavings$ with the move by $tmpSavings, newSavings, waitDiff, \alpha$ and $newWaits$, if costs are less with the move, make the move, and recalculate the pickup times.
STEP 13. Establish the tabu list with the principle of 1) if 25 iterations have passed, begin to add Tabu list, 2) after $tabuMax$ iterations, candidates leave the tabu list.
STEP 14. Find the objective of new solution, calculate the $localObjective$ with the objective function of $localObjective = \alpha * newWaits + cost$.
STEP 15. Check if the $newWaits$ of violation value of constraints, if it is 0, modify the punishment coefficient $\alpha = \alpha/(\sigma + 1)$, otherwise, $\alpha = \alpha * (\sigma + 1)$.
STEP 16. Check the best objective value, if the objective value is larger than localObjective, update the $Route$ and $requestMat$ and assign the $objective$ with $localObjective$ value.
STEP 17. Increment the iteration count and repeat the tabu search procedure.
STEP 18. Calculate the $objective$ and $sumDist$, display $objectives, newWaits$ and α, then plot routes.
STEP 19. Output $objective$ value, $cost$ value, $newWaits$ value, α, $Routes$ matrix, $requestMat$ matrix, figure of routes.

Function of $Move$
Variables: $Routes, r1^*, r2^*, p1^*, p2^*, j1^*, j2^*$
Make the move:
STEP 1. Remove customer j^* from route $r1$.
STEP 2. Add customer j^* to position $p2^*$ in route $r2$.
STEP 3. Shift the end of the route by 1 position.
STEP 4. Insert j^* into position $p2^*$

Function $findTime$
Variables: $pickupTime, idle, distances, Routes, requestMat, V, r$
Calculate the pickuptime of each request:
STEP 1. Set initial $pickupTime$ of request r as 0, $idle$ of request r with 0, here r is the ID of request.
STEP 2. Start calculation from the second node of each route (first node is depot), with the interval of 2 ,until the second to last node (last node is depot).
STEP 3. Calculate $pickupTime$ by $idle$, $pickupTime$ and $distances$ between sequential order of pairwise nodes which vehicle traverse the route with each request, including links of trip, relocation and idle waiting for customer.

Function $waitCalculator$
Variables: $car, Routes, distance, requestMat, V, pickupTime$
Calculate the waitDiff:
STEP 1. Set waitDiff value 0.
STEP 2. Find the deviation between actual pickupTime and latest pickup time of request.
STEP 3. Accumulated assign the value to $waitDiff$.

5.2 K-Means - Tabu search

K-Means with tabu search is proposed to split the initial problem size into smaller subproblems and reduce the computation time with multi-threading parallel calculation. The detailed process is described below:

STEP 1. Reform the dataset of original request matrix into a $R_i(Pt_i, Dt_i)$ attributes matrix X which represents the requests as temporal feature data points.
STEP 2. Input the dataset of X K-Means instructions, apply an iterative refinement technique and assign each x to the cluster whose mean has the least city block distance, and intuitively get the nearest mean.
STEP 3. Multi-threading parallel calculation of the absolute differences sum and each centroid is the component-wise median of the points in that cluster and aims to minimize sum in cluster with the nearest mean according to objective function of $D(x, \lambda) = argmin_X \sum_{x \in S_i} |x_i - \lambda_i|$.
STEP 4. Calculate the new means to be the set of centroids k, and assign the observations in the new clusters.
STEP 5. Repeat step 3, 4 and get the optimal solution of k cluster groups with corresponding centroids.
STEP 6. Assign the k groups requests clustering results to each corresponding computing node.
STEP 7. Each computing node execute the Tabu search algorithm independently with the corresponding assigned requests group.
STEP 8. Recieve the routes from all computing nodes, join all the routes to obtain a final solution.

5.3 K-Medoids - Tabu search

K-Medoids algorithm with tabu search is also applied to decompose the problem size and increase the calculation efficiency by utilizing parallel computing methods. The detailed process is described below:

STEP 1. Input dataset of original request matrix and distance matrix of travel time between nodes.
STEP 2. Parallel compute the distance matrix according to the pairwise requests distance objective function $D(R_i, R_j)$ mentioned before.
STEP 3. Apply the distance matrix *requestdistanceMat* to K-Medoids algorithm package in R, get the result of k request groups and a set of medoids (Centers).
STEP 4. Assign the k request groups results to each corresponding computing node.
STEP 5. Each computing node execute the Tabu search algorithm independently with the corresponding assigned requests group (same procedure as Tabu algorithm aforementioned).

STEP 6. Receive the routes from all computing nodes, join together and obtain the final solution.

Chapter 6

Computational experiments

This chapter consists of four major sections. First section start with the introduction of the experiment goals, applied algorithms, computation environments and the cases of RACS routing problems to be studied. Section 2 utilize a preliminary test to prove the effectiveness of the algorithms. In section 3 empirical study, execute the algorithms with one case of each small scale and large scale problems. For each problem 3 tests has been implemented, with all the results, stability of each algorithm is evaluated. Then analysis the optimization results and compare the performance of algorithms according to solution quality and computation time. Last section shows the discussion based on the analysis of results. In the chapter following research questions are addressed
What are the performance and case study results of the algorithms?

6.1 Experiments setting

In order to efficiently find the optimal feasible solutions for the optimization problem of RACS routing, we proposed three algorithms into the solution procedure and try to prove their feasibility and optimization effectiveness with real life case studies. Since our problem is new, there are no benchmark instances could be compare to. Therefore, in order to illustrate the efficiency of each algorithm, we apply them to all cases with diverse size of demand and then, by comparing the performance of algorithms, determine the most appropriate algorithm fit for each corresponding size of problem. Tabu search, K-Means Tabu search (KMN-Tabu) and K-Medoids Tabu search (KMD-Tabu) are applied into the empirical optimization problems include one preliminary test problem and two real life problems with small and large scale.

The computation environment for all cases are implemented under R and Matlab with Intel core i5-7400CPU 3.00 GHZ computer, which can execute multi-threads parallel computation with maximum 4 cores. For the empirical study, our instances including two networks with different size problems, which comprise of a small size of *Siouxfall* network and a large size of *NewYork* Taxi network. The request dataset of each problem is either randomly created or extracted from historical instances of demand. All the cases settings would be showed in the parameter table. During the implementation, vehicle routing and request dispatching matrix will be obtained according to the objective which denotes minimum travel cost and waiting time. Moreover, computation time and result comparison are analysis afterwards.

6.2 Algorithm effectiveness analysis

Aiming at providing evidences concerning the optimization effectiveness and the optimality of solutions with the algorithms applied in the problem, we proposed a simple network (referred to as Testnetwork), which is randomly created with 11 nodes. It should be noted here, that the mixed integer problem is not presented for any case, mainly based on the objective of this thesis which is the exploration of an optimization algorithm for large scale networks.

First of all, we initial the network based on the requests pickup and drop off nodes, coordinates which could be seen as zones or area centers. Then we create travel time/cost matrix based on the assumption that travel time on each arc during the planning time are known under the perfect information. A fleet of 8 homogeneous AVs and one depot which is denoted as node 11 are provided in the system. Since the RACS is a reservation based system, requests are randomly created with a small set of data comprised 20 instances according to the planning time period within 300 min (5 hours) time units.

Problem settings and input parameters are presented in Table 6.1.

Table 6.1: Test network problem size and parameter setting

Problem size and parameters						
Network	*Requests*	*Vehicles*	*Travel Cost*	*alpha*	*sigma*	*Max Iterations*
11	20	8	Distances matrix	1	0.5	10000

6.2.1 Tabu search algorithm

We first applied the Tabu algorithm directly to the test problem. For each test a high number of iterations is explored with results for each test to logged every 100 iterations, and each experiment involved a different number of iterations. This analysis was performed in three tests. After maximum 10000 iterations execution of the program, solutions are plotted composing of three tests, which could be seen in Figure 6.1.

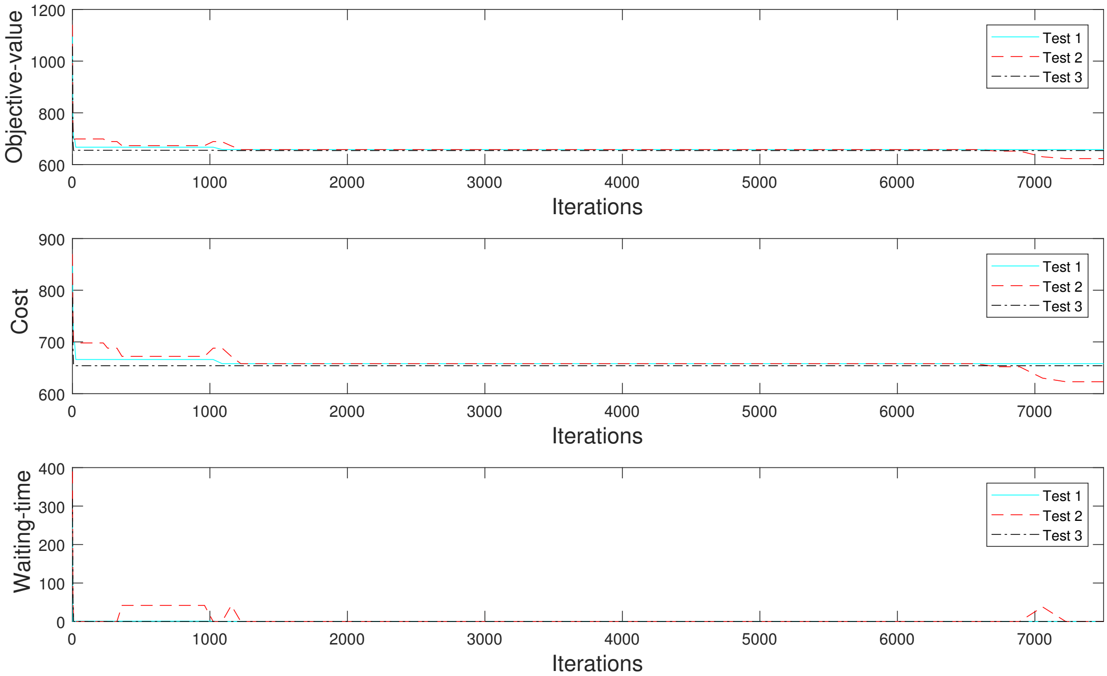

Figure 6.1: Testnetwork Tabu search solution

Optimization stability analysis

The results that are presented in Figure 6.1 and those from Table 6.2 are used to derive conclusions concerning the performance and sensitivity of the Tabu Algorithm. It is clear that there is a obvious decreasing of objective value, cost value and waiting time value. This indicates the optimization effects that the Tabu algorithm has. As it is clearly presented, the Tabu search manages to get away from a local optimum after a large number of iterations and conclude to a lower objective function. The variation between the three test results and mean value is relatively small as well as standard deviations. For the objective value and cost, the biggest interval is 22 mins, which is 3%, and waiting time violation is zero. Therefore, the best results based on maximum iteration 10000, which could be found by Tabu search algorithm is stable. It has to be noticed that Tabu search could not guarantee the global optimum solution, the obtained best results only based on the iterations time, which may still be a local optimum.

Table 6.2: Tabu search algorithm solutions of Testnetwork with three tests

Best Result	Mean	Standard dev.	Test 1	Test 2	Test 3
Objective value	645	15.64	623	654	658
Cost	645	15.64	623	654	658
Waiting time violation	0	0	0	0	0

Another interesting result presented in Figure 6.1, refers to the interval between 800 -1000 and 7000 - 7200 iterations, where each value has fluctuation because of the diversification strategy which explores more neighborhood solution space and exchange of the requests between routes, it leads to a lower travel cost, but cause more waiting time violation.

6.2.2 K-Means-Tabu search algorithm (KMN-Tabu)

K-Means-Tabu search is also applied to the Testnetwork with maximum iteration of 10000, which decompose the initial problem size into smaller sub-problems, and parallel execute the Tabu search algorithm. Here we give a presetting of 4 clusters which indicates that all the requests will be divided into 4 groups and each of them will be served by 2 AVs. Parts of the three tests solutions are plotted in Figure 6.2.

Optimization stability analysis

Table 6.3: KMN-Tabu algorithm solutions of Testnetwork with three tests

Best Result	Mean	Standard dev.	Test 1	Test 2	Test 3
Objective value	787.3	5.18	791.0	791.0	780.0
Cost	754.7	4.7	758.0	758.0	748.0
Waiting time violation	32.7	0.47	33.0	33.0	32.0

From the sensitivity curves of objective value, cost and waiting time in Figure 6.2, the solution optimization progress could be found based on the value decreasing of each indicator, which also proves the algorithm effectiveness. Meanwhile, in table 6.3, the maximum deviation of objective value between mean and three tests is about 0.4%, and

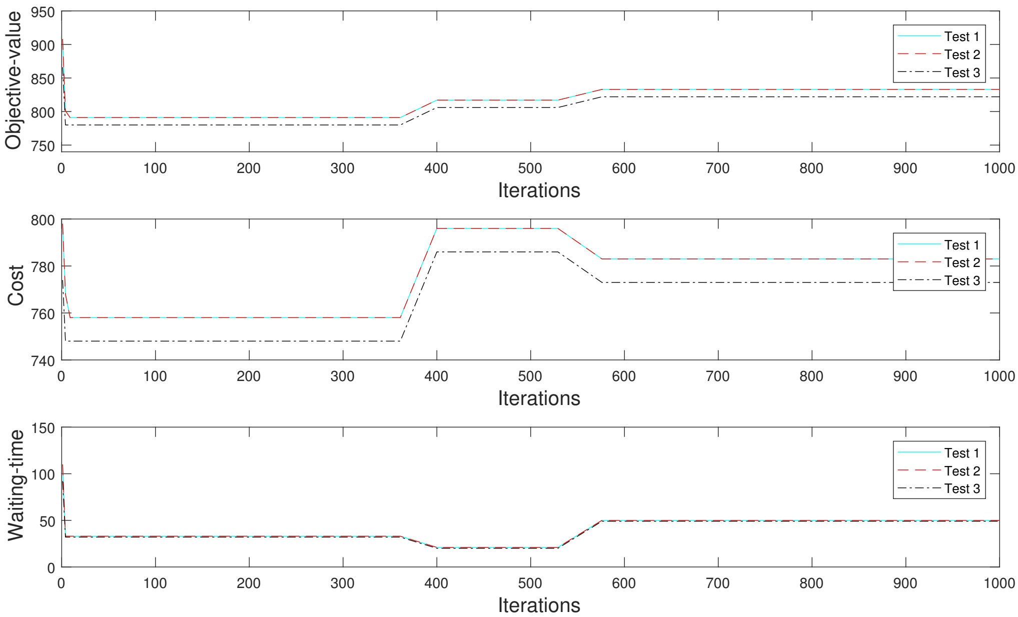

Figure 6.2: Testnetwork K-means - Tabu search solution

standard deviation is 5.18, therefore the algorithm could be denotes as stable. Nevertheless, some value fluctuations also could be seen from iterations 520 to 580 in both cost value and waiting time value curves. During the interval of iterations, cost value is reducing by inserting requests between routes, however, incurs violation of time window which raises the waiting time value. Conversely, same pattern happened during the iteration 350 to 400 as well. It has to be noticed that if a local optimal is reached in Tabu search, the algorithm will start receiving less optimal solutions to escape the local optimal and attempt to find better solution to the problem by exploring larger search space. Therefore, the curve behave increasing. In the end of maximum iterations, the solution with the lowest objective function value is kept.

6.2.3 K-Medoids-Tabu search algorithm (KMD-Tabu)

Similar to KMN-Tabu, KMD-Tabu is also explored on a set 4 clusters of all request and divided them into 4 groups with each 2 AVs. Parts of the three tests solutions with maximum 10000 iterations are plotted in Figure 6.3.

Optimization stability analysis

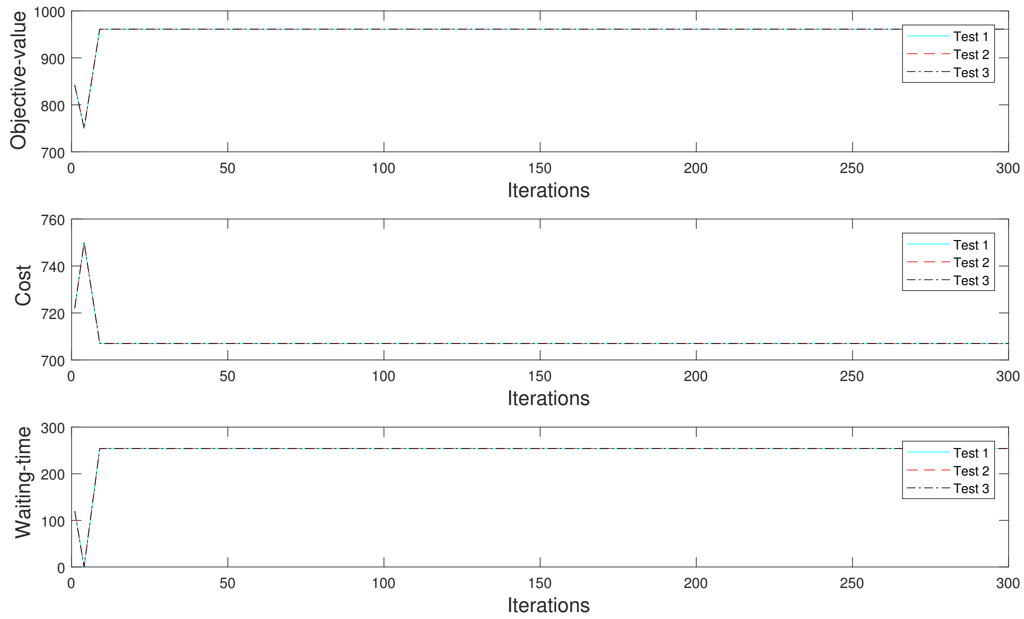

Figure 6.3: Testnetwork K-Medoids-Tabu solution

Table 6.4: KMD-Tabu algorithm solutions of Testnetwork with three tests

Best Result	Mean	Standard dev.	Test 1	Test 2	Test 3
Objective value	750.0	0.0	750.0	750.0	750.0
Cost	750.0	0.0	750.0	750.0	750.0
Waiting time violation	0.0	0.0	0.0	0.0	0.0

According to value of objective, cost and waiting time in figure 6.3, the algorithm found minimum solution in 9th iteration, then start to escape the local optimal, the sensitivity curves also show that the algorithm is effective for optimization. Furthermore, from table 6.4, there is no deviation between three tests and mean value, standard deviation is zero, where stability of the solution is proved.

6.2.4 Algorithm performance analysis

Based on the solution from table 6.2, 6.3 and 6.4 and computation cost comparison of the three algorithms in figure 6.5, we could find some features of their performance as following:

1. The results of all algorithms showed a optimization process, which proved the

effectiveness for RACS routing optimization. However, it has to be notice that global optimal solution could not be guaranteed with all three algorithms which are based on Tabu search.

2. Tabu search phase aims to find the better optimal solution within a number of iterations, when come to the local optimal, it will start to allow the moves deteriorate current solutions to avoid stuck in poor-scoring area. Therefore, we could find the fluctuations in the curves. Since cost and waiting time value are related to each other, when the cost optimization executed during the Tabu search phase, the reinsertion of requests will exchange links within routes, which may cause new violation of time window, thus, the waiting time will increase, and vice versa. For example, from iteration 520 to 580 in KMN-Tabu algorithm (see in figure 6.2), when cost optimization start to diversify and explore more search space, incurs violation of time window which cause the waiting time cost raising.

3. The results of three algorithm are all stable, nevertheless, from three algorithm mean value comparison (see in figure 6.4), it could be found that Tabu algorithm obtains higher quality than K-Means-Tabu and K-Medoids-Tabu. The main reason of inferior quality of solutions is that after split the requests with 4 groups, whole optimization problem is decomposed into 4 sub-problems with smaller size, correspondingly, each sub-problem optimization would have relatively smaller search space for exploration and the neighborhood are limited, with iterations increasing, it is hard to find better feasible solutions.

4. From the computation time/cost perspective, Tabu spend much more time than other two algorithm(see in figure 6.5). However, if the problem is relatively small and the actual value of the cost is low, then Tabu algorithm could be acceptable.

5. KMD-Tabu solutions shows its stability advantage, which generate a solution that is considered to be more robust than KMN-Tabu algorithm.

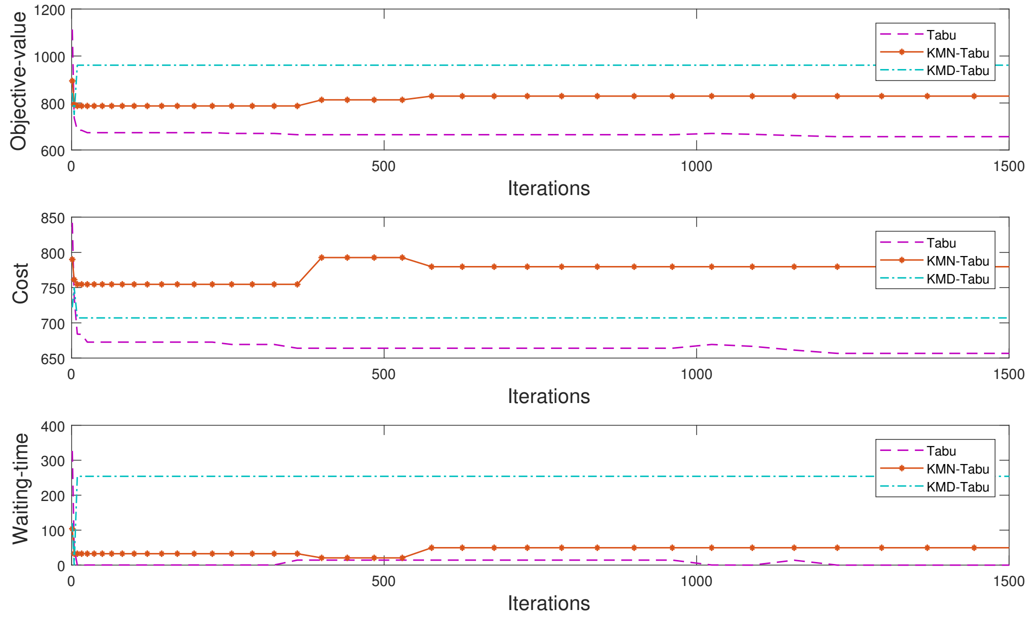

Figure 6.4: Testnetwork three algorithms mean value comparison

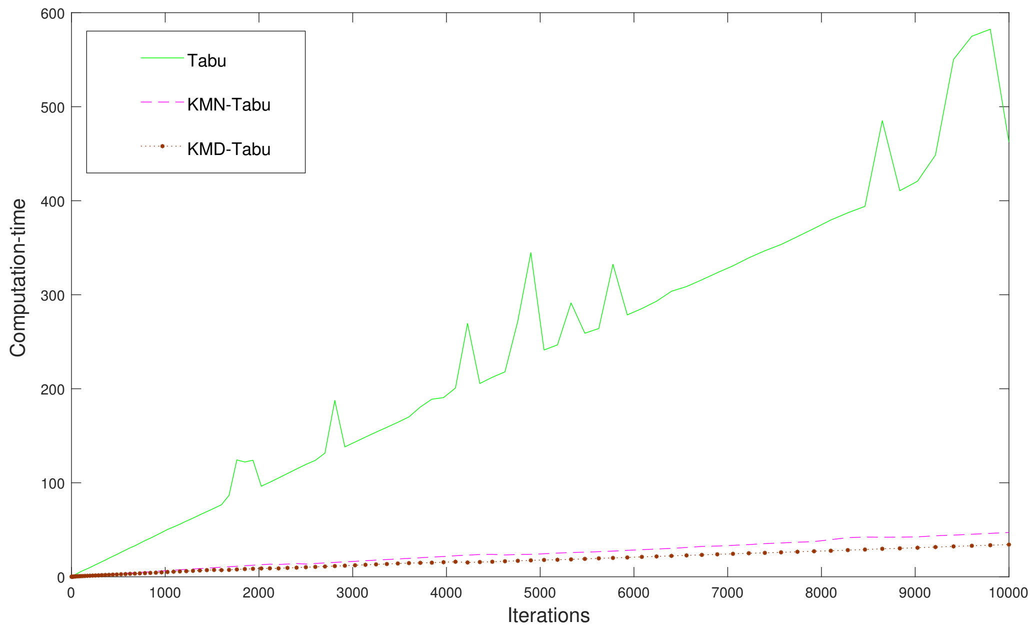

Figure 6.5: Testnetwork compuation cost comparison

6.3 Empirical Study

6.3.1 Small scale case: Siouxfalls Network

6.3.1.1 Case background

The small scale problem of Siouxfalls composes of 24 nodes in the network. We set the system with node 11 as depot, 40 AVs will provide services to 100 requests within planning time of 300 min (5 hours). Travel time/Cost matrix are created based on the information from the network [Bbstabler, 2018]. The requests are randomly created using a 15min interval time window. Parameters chosen for this experiment could be seen in table 6.5.

Table 6.5: Siouxfalls network problem size and parameter setting

Problem size and parameters							
Network	*Requests*	*Vehicles*	*Travel Cost*	*alpha*	*sigma*	*Iterations*	*Time-window interval*
24	100	40	Dist-Matrix	1	0.0005	max 5700	15 min

6.3.1.2 Tabu search algorithm

Firstly, Tabu algorithm is applied to the problem and performs a sensitivity analysis. For each test, 76 results will be logged following an exponential growth on the number of iterations.

Based on the three test results, we use the mean value to evaluate each test performance. After maximum 5700 iterations execution of the program, solutions are plot composing of three tests and their mean value, which could be seen in Figure 6.6 ,6.7 and 6.8.

Optimization stability analysis

Table 6.6: Tabu search algorithm solutions of Siouxfalls network with three tests

Best Result	**Mean**	**Standard dev.**	**Test 1**	**Test 2**	**Test 3**
Objective value	4113.7	12.28	4118	4093	4091
Cost	3961.3	31.56	3960	4022	4031
Waiting time violation	96	43.83	158	71	60

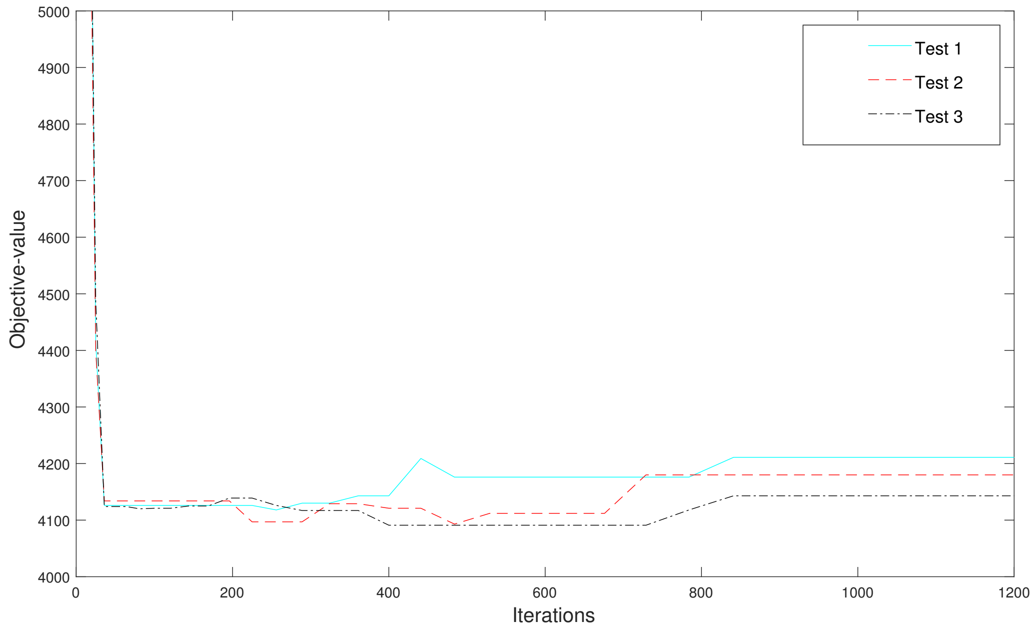

Figure 6.6: Siouxfalls Network Tabu search objective value solution

Table 6.6 and sensitivity curves in figure 6.6, 6.7 and 6.8 provide an overview of the best results of Tabu algorithm solution. Specifically:

1. For Objective value the biggest interval is 22.7 mins, which is only 0.5% and standard deviation is 12.28.

2. Largest deviation in the cost is 1% and standard deviation is 31.56.

3. Waiting time value has obvious fluctuation with 64% and standard deviation is 43.83.

Since our objective is to minimize the total value of cost and waiting time, based on this priciple objective value can be our main factor to judge the stability. Therefore, solutions of Tabu search algorithm can be determined to be stable.

From figure 6.7 and 6.8, we could find out that the fluctuation of cost and waiting time value are related, from sections of iterations 200 to 400 and 650 to 800, waiting time value decreasing with the increase of cost value, this situation happens when exchange links between routes, limited the time window, however increases the travel cost.

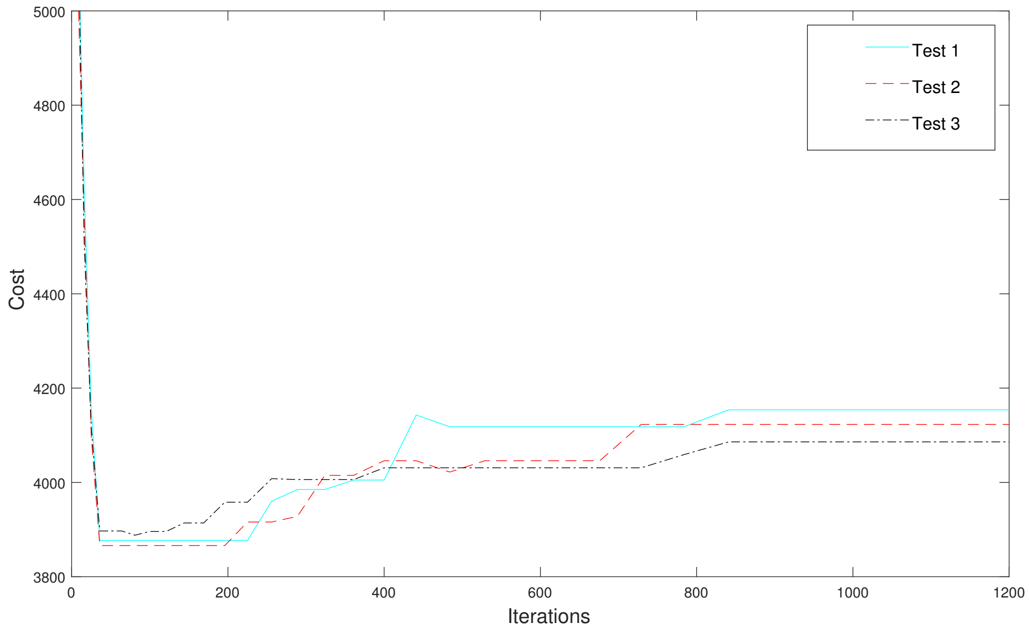

Figure 6.7: Siouxfalls Network Tabu search cost solution

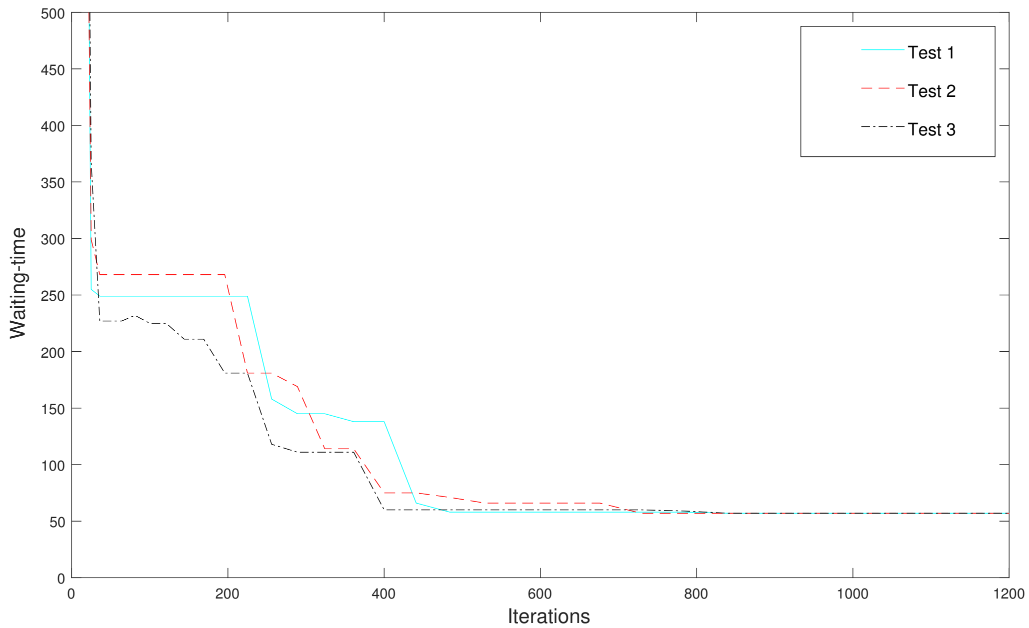

Figure 6.8: Siouxfalls Network Tabu search waiting time solution

6.3.1.3 K-Means-Tabu search algorithm (KMN-Tabu)

With KMN-Tabu algorithm, the initial problem has been decomposed into 4 clusters, the requests of each cluster will be served by 10 AVs. All the three tests solutions are plot in

Figure 6.9, 6.10 and 6.11.

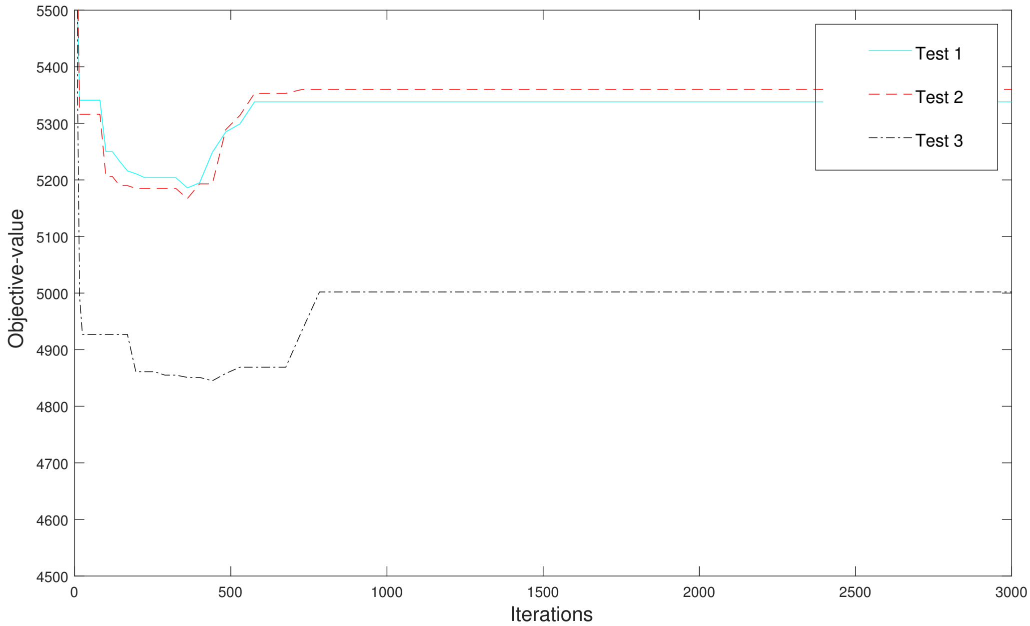

Figure 6.9: Siouxfalls Network K-Means-Tabu search objective value solution

Optimization stability analysis

Table 6.7: KMN-Tabu algorithm solutions of Siouxfalls network with three tests

Best Result	Mean	Standard dev.	Test 1	Test 2	Test 3
Objective value	5068	156.46	5186	5167	4845
Cost	4398.7	239.73	4252	4206	4736
Waiting time violation	669.3	395.42	934	961	109

Figures 6.9, 6.10 and 6.11 and table 6.7 give an overview of the maximum deviation between mean value and three tests. Specifically:

1. Objective value varies from 2% - 4%, standard deviation is 156.46.
2. Cost value varies from 4% - 7.7%, standard deviation is 239.73.
3. Waiting time value varies from 39% - 83% , standard deviation is 395.42.

From the objective value variation between three tests, since the standard deviation is too large, we could conclude that the solution of KMN-Tabu algorithm is not stable. However this should be further evaluated by more tests.

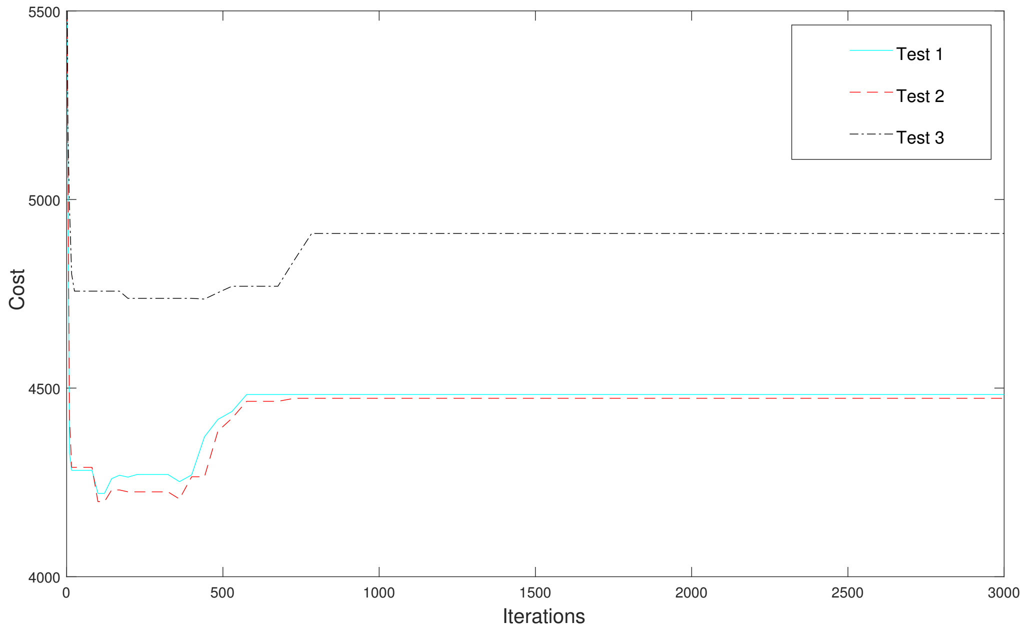

Figure 6.10: Siouxfalls Network K-Means-Tabu search cost solution

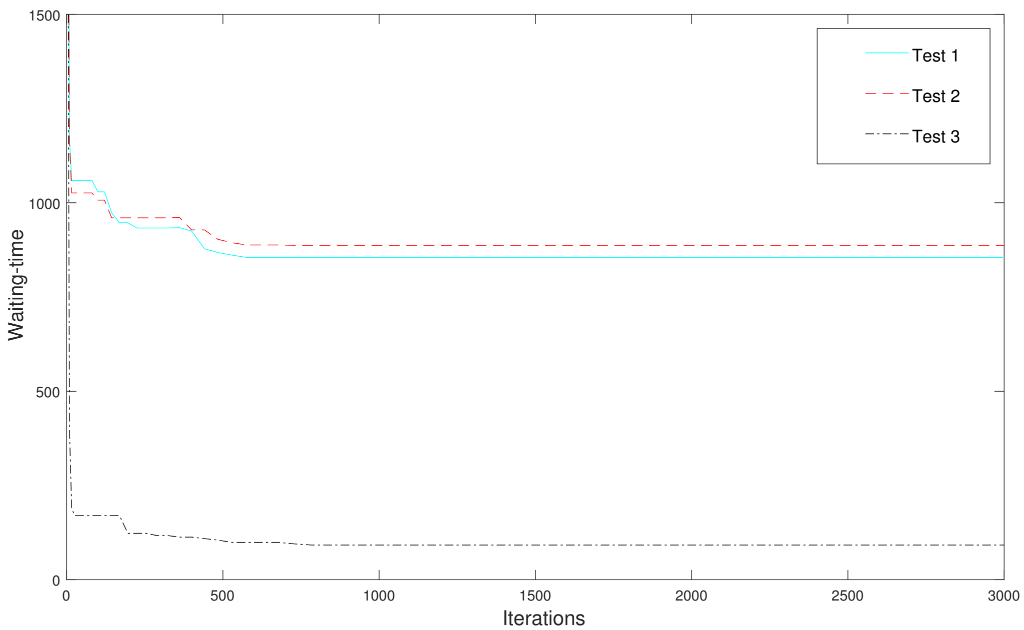

Figure 6.11: Siouxfalls Network K-Means-Tabu search waiting time solution

6.3.1.4 K-Medoids-Tabu search algorithm (KMD-Tabu)

The same settings were used also for the KMN-Tabu. All request are divided into 4 groups with each 10 AVs provides the services. All three tests solutions are plot in Figure 6.12, 6.13 and 6.14.

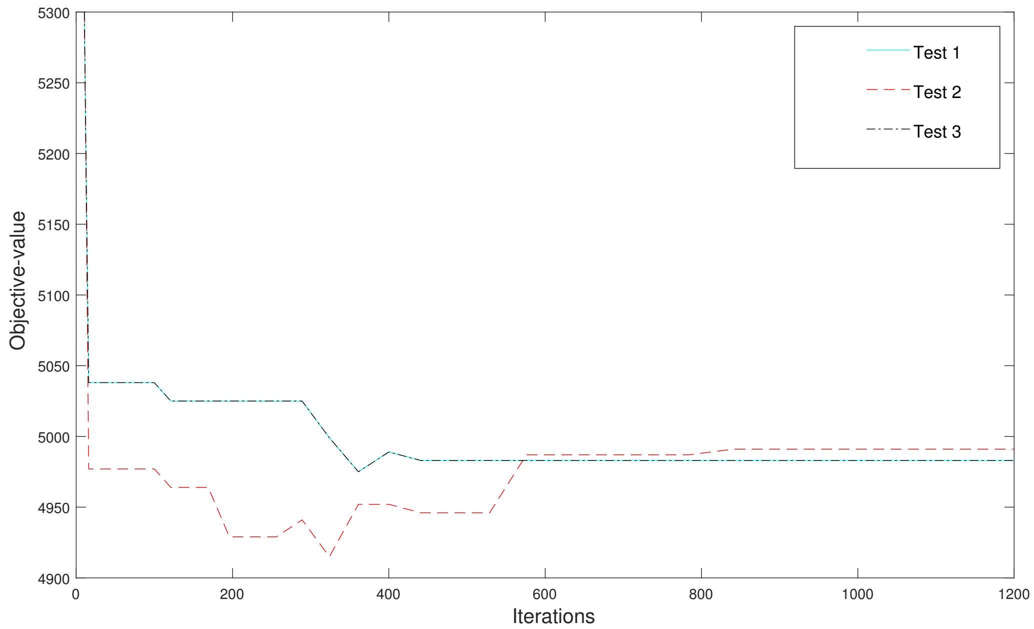

Figure 6.12: Siouxfalls Network K-Medoids-Tabu search objective value solution

Optimization stability analysis

Table 6.8: KMD-Tabu algorithm solutions of Siouxfalls network with three tests

Best Result	Mean	Standard dev.	Test 1	Test 2	Test 3
Objective value	4970.6	28.28	4975	4915	4975
Cost	4866.3	48.08	4869	4767	4869
Waiting time violation	104.3	19.79	106	148	106

From figures 6.12, 6.13 and 6.14 and table 6.8, the standard deviation and mean value among three tests are concluded. Specifically:

1. Objective value varies from 0.08% to 1%, standard deviation is 28.28 min.
2. Cost value varies from from 0.05% to 2%, standard deviation is 48.08 min.
3. Waiting time value varies from 2% to 41%, standard deviation is 19.79 min.

Based on the variation range of objective value, solutions of KMD-Tabu search algorithm can be determined to be stable. The fluctuation between iterations 120 to 600 represents the process of exchange links between routes, where time window violation decreasing with cost increasing.

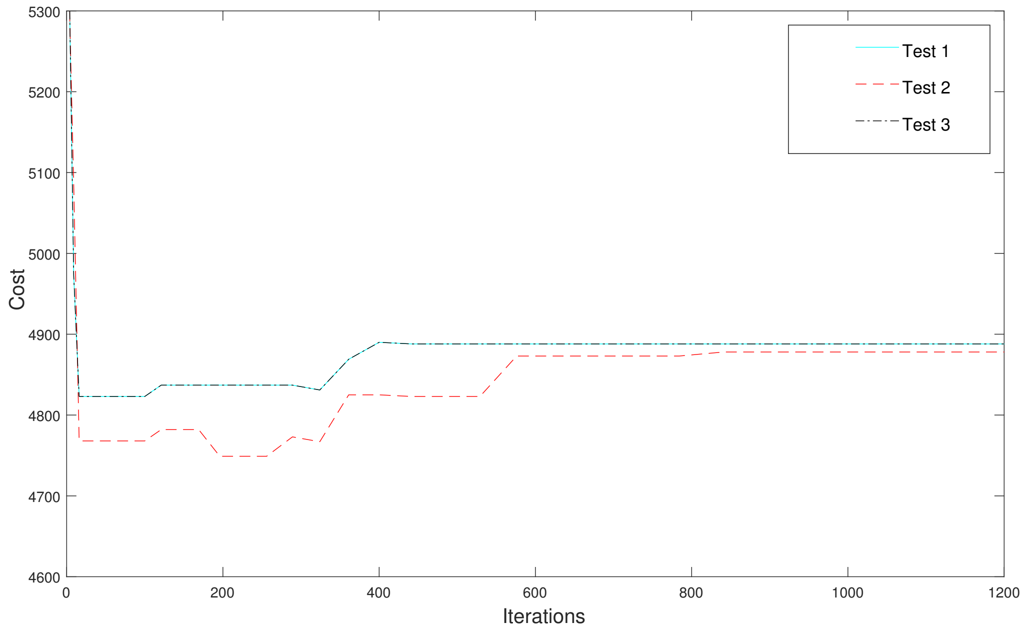

Figure 6.13: Siouxfalls Network K-Medoids-Tabu search cost solution

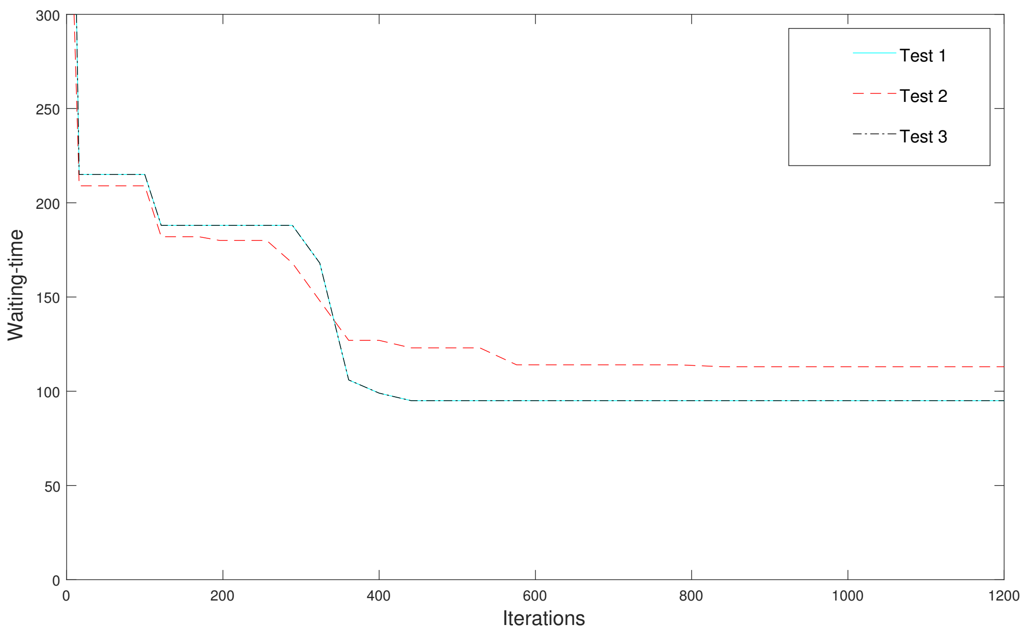

Figure 6.14: Siouxfalls Network K-Medoids-Tabu search waiting time solution

6.3.1.5 Performance Comparison

Solution quality analysis

By comparing with the best value of each test in table 6.9 and mean values of three algorithms in figure 6.15, we could find out that Tabu search obtains the highest quality solution among the algorithms.

Table 6.9: Best solutions of three algorithm for small scale problem

Best Result	Objective value	Cost	Waiting time	Iteration	Computation time (s)
Tabu	4091.0	4031.0	60.0	400.0	545.8
KMN-tabu	4845.0	4736.0	109.0	441.0	77.1
KMD-Tabu	4915.0	4767.0	148.0	324.0	86.3

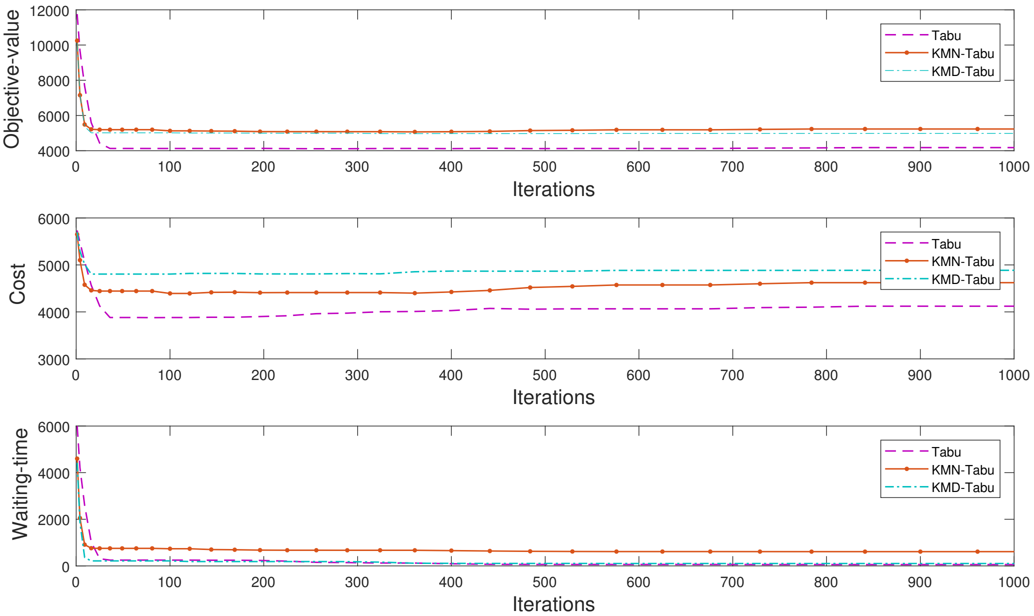

Figure 6.15: Siouxfalls Network three algorithms mean value

Since the clustering method has divided the initial problem into 4 group, the search space is relatively limited, therefore solution quality of KMN-Tabu and KMD-Tabu is lower than Tabu algorithm. However, the computation time for Tabu search is up to 5 times higher than the other two algorithm.

Computation cost analysis

By comparing the best results iteration computation time and sensitively analysis the behavior in figure 6.16, it can be concluded that Tabu search spend higher computation cost than KMN-Tabu and KMD-Tabu, however, since the problem size is small, the computation time of Tabu algorithm is still acceptable.

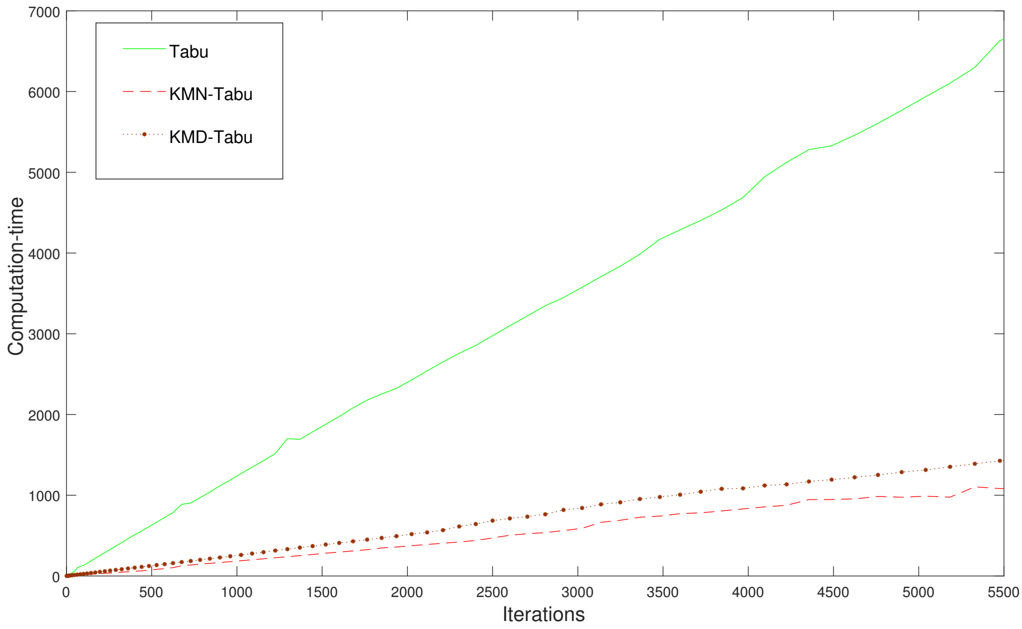

Figure 6.16: Siouxfalls Network computation cost comparison

6.3.1.6 Result discussion

Based on the above analysis of solution quality and computation time comparison, presented in Figure 6.17, the following conclusion could be made for *Siouxfall* network problem:

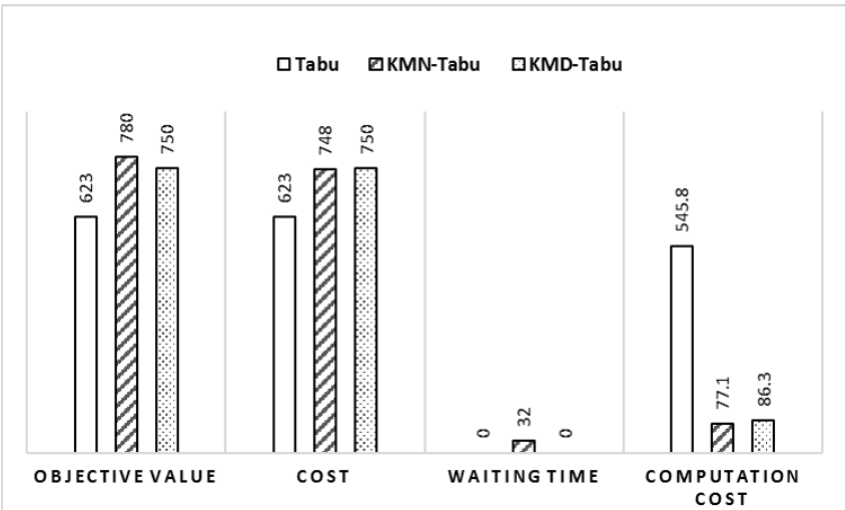

Figure 6.17: Small scale problem of Siouxfall results conclusion

1. The results of Tabu and KMD-Tabu algorithms are stable, nevertheless, Tabu search obtains higher quality than KMD-Tabu.
2. From the computation time/cost perspective, Tabu search spend much more cost than other two algorithm. However, since the size of the problem is small and the time value still can be accepted.
3. K-medoids-Tabu solutions is more robust than K-means algorithm.

According to the algorithm performance analysis, Tabu search seems to be more appropriate for small scale RACS problems as it results to better quality results with an acceptable computation cost.

6.3.2 Large Scale problems: NewYork taxi Network

6.3.2.1 Case background

In order to test our model effectiveness and efficiency with real life problem which contain large size of requests, we used the green taxi data from TLC Trip Record Data of *NewYork* city [NYCTLC, 2018] in January 2017, and particularly 3 hours demand data of 9:00 - 12:00 on February 1st with 3968 requests. Since year 2015, *NewYork* TLC authority no longer provides the coordinates for each trip instead with the zone locations, here we apply the cited dataset with network of 263 zones nodes which represent the taxi zones for the green taxi operation (see in figure 6.18) and designate node 11 as depot. From the shape file provided by the TLC, within each zone we settled one centroid location coordinates to represent the pickup and drop-off stations for each trip origin and destination.

For the demand model, the initial cited dataset only including pickup, drop off locations and departure time of each request. In order to accompany with our model, we modified the dataset matrix by adding the customer tolerance waiting time as 15 minutes. We take both time variables to obtain the time window of earliest pickup and latest pickup time represented by departure and plus tolerance time.

Furthermore, we use the coordinates and city block distance metric method to get the cost model. To simplify the cost model, we assume that during the planning time, vehicles speed are the same on each path with 40 km/h.

The parameters used are presented in Table 6.10:

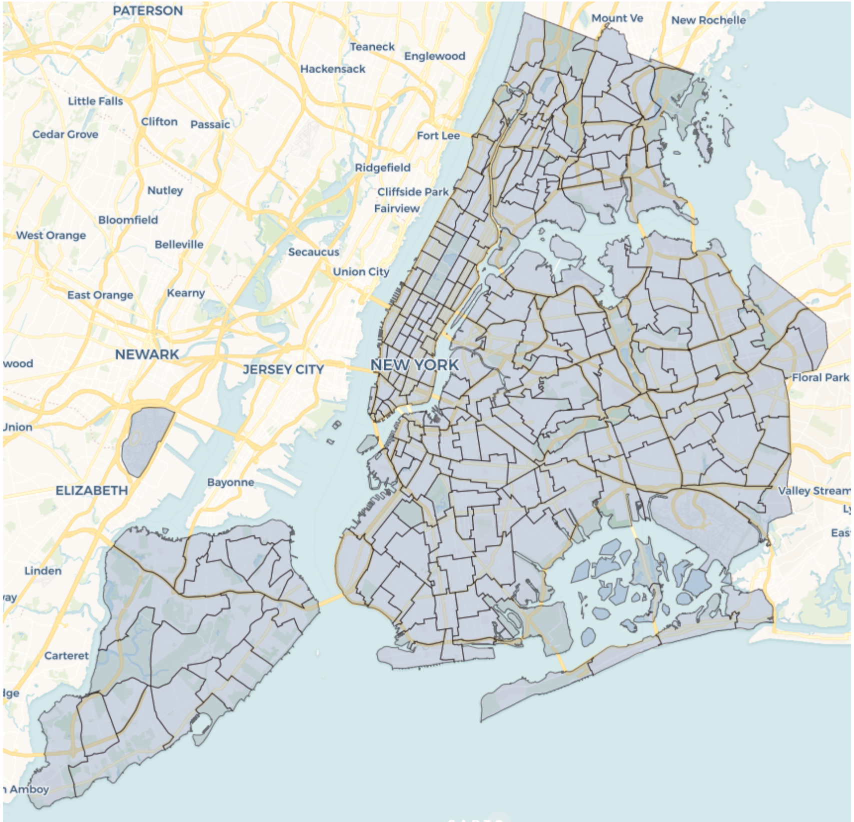

Figure 6.18: NewYork taxi zones and network

Table 6.10: NewYork taxi network problem size and parameter setting

Problem size and parameters							
Network	*Requests*	*Vehicles*	*Travel Cost*	*alpha*	*sigma*	*Iterations*	*Time-window interval*
263	3986	800	Dist-Matrix	1	0.0005	max 3000	15 min

6.3.2.2 Performance Comparison

Before the formal test, we have preformed several pre-runs for the Tabu algorithm where we found that the computation costs of this method is extremely high, and it is not applicable to adopt in large scale of problems solution. Hence, we only applied KMN-Tabu and KMD-Tabu algorithm into the solution process. We test KMN-Tabu algorithm once with maximum 3000 iterations, KMD-Tabu with maximum 700 iterations due to the time limit. Hence, we will plot the sensitivity curves of their performance based on 700 iterations with 7 results of each. With KMN-Tabu and KMD-Tabu algorithms, the initial problem has been decomposed into 100 clusters, where the requests of each cluster will be served by 8 AVs. Both algorithm solutions are plot in Figure 6.19, 6.20 and 6.21.

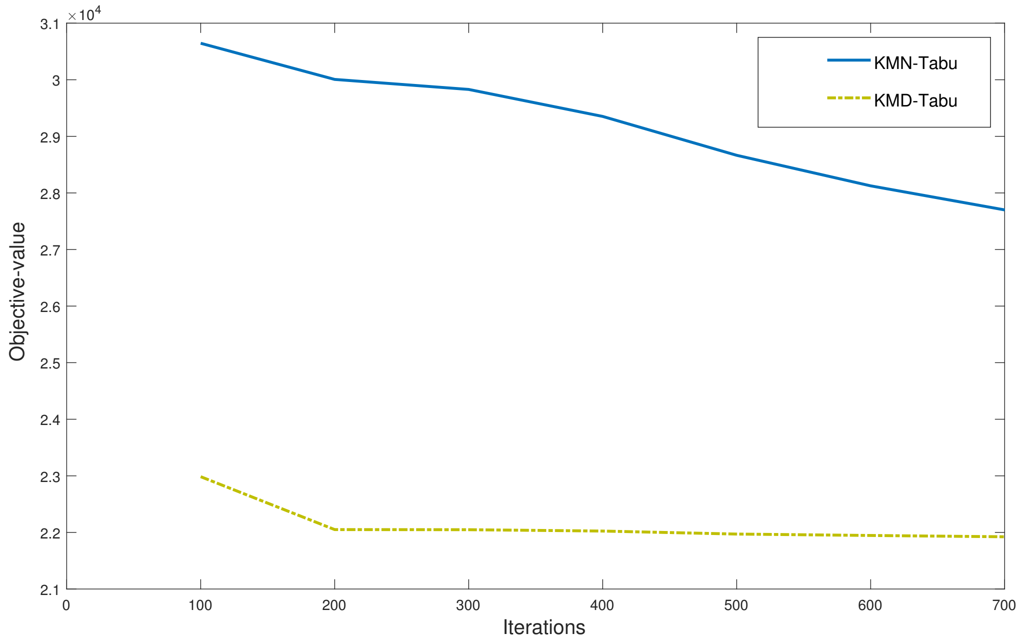

Figure 6.19: NewYork taxi Network KMN-Tabu and KMD-Tabu search objective value solution

Solution quality analysis

From figure 6.19, 6.20 and 6.21, we could see obviously optimization processes of both algorithms. By comparing with the best value of each method in table 6.11, we could find out that KMD-Tabu obtains the higher quality solution than KMN-Tabu algorithm with less iterations.

Table 6.11: Best solutions of three algorithm for small scale problem

Best Result	Objective value	Cost	Waiting time	Iteration	Computation time (s)
KMN-Tabu	27042	25815	1227	3000	1281.5
KMD-Tabu	21923	21896	27	700	2643.25

Computation cost analysis

By comparing the best results iteration computation time and sensitively analysis the behavior in figure 6.22, it could be concluded that KMD-Tabu requires longer computation time with same iteration setting.

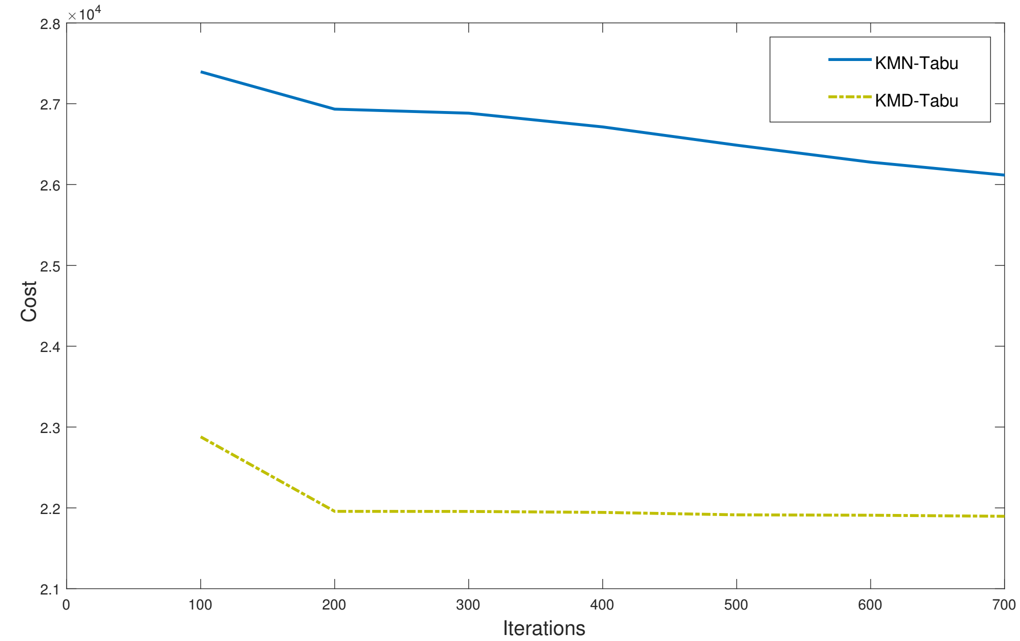

Figure 6.20: NewYork taxi Network KMN-Tabu and KMD-Tabu search cost solution

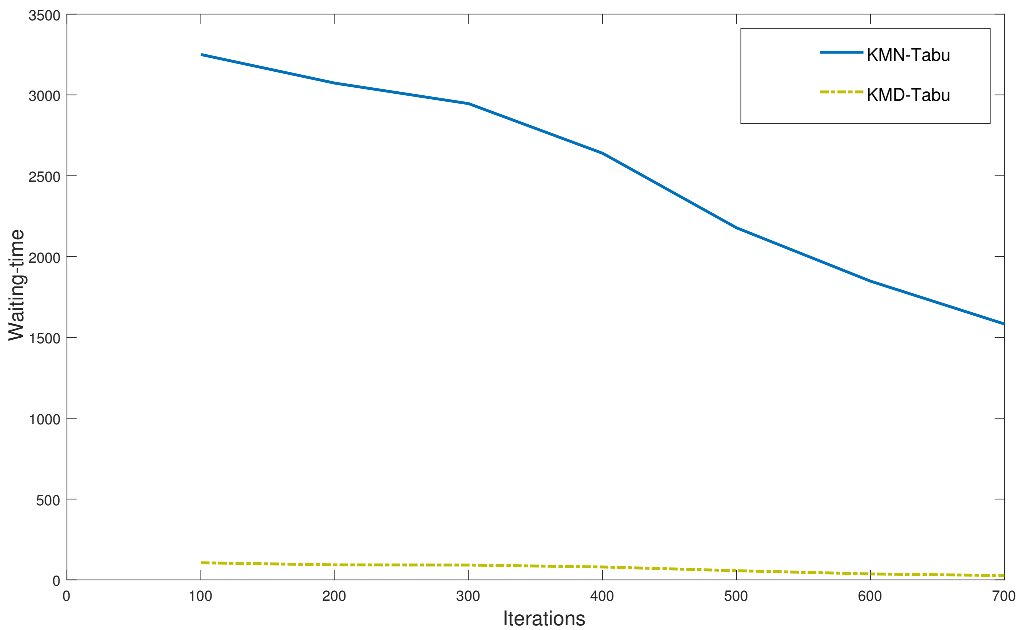

Figure 6.21: NewYork taxi Network KMN-Tabu and KMD-Tabu search waiting time solution

6.3.2.3 Result discussion

Based on the above analysis with solution quality and computation time comparison, the following conclusion could be made for *NewYork* taxi network problem.

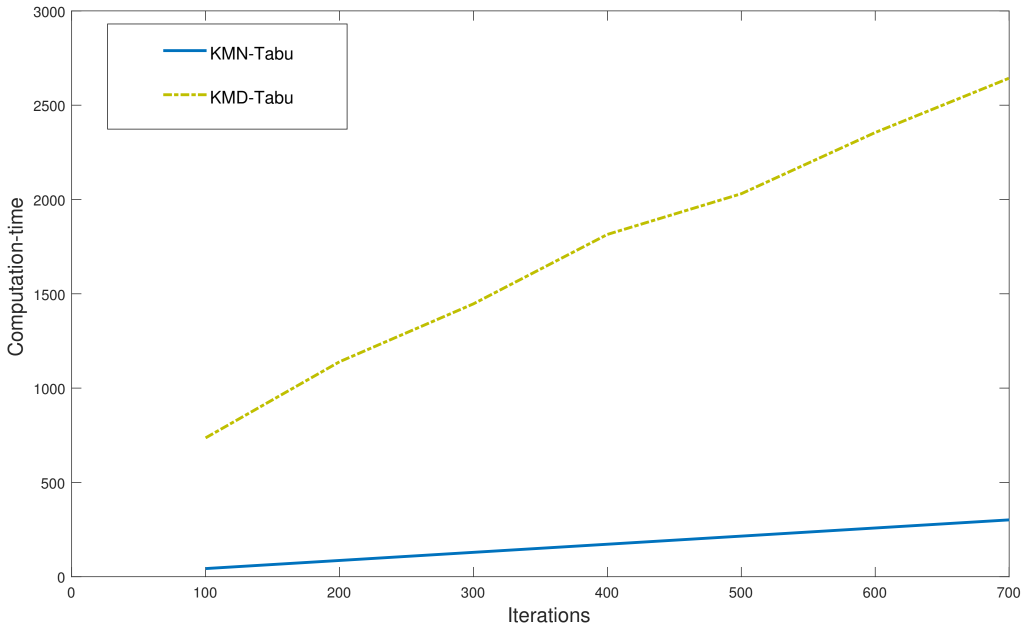

Figure 6.22: NewYork taxi Network computation cost comparison

1. KMD-Tabu obtains higher quality solutions than KMN-Tabu algorithm with less iterations calculation.
2. Compare to K-Means method, K-Medoids algorithm is more effective on clustering with requests.
3. KMD-Tabu spend much higher computation cost than KMN-Tabu with same iterations. However, since the problem size is relatively large, corresponding, the computation cost of KMD-Tabu could be acceptable.

According to the algorithm performance analysis, KMD-Tabu algorithm is more suitable for the large scale of RACS routing problem which gives higher quality solutions.

6.4 Summary and discussion

Based on the computation experiment results of three networks problems including preliminary, small scale and large scale with Tabu, KMN-Tabu and KMD-Tabu algorithms we reach the following conclusions:

1. All three problem has proved the consistency of Tabu algorithm optimization

effectiveness. However, the global optimal solution is not guaranteed.

2. The solution optimality of Tabu search before the clustering is better than after, since K-Means and K-Medoids algorithm decompose the problem size where reduce the searching and exploration space for each sub problems, which scarifies the optimality for reducing computation time. Therefore, among all three methods, Tabu algorithm provides the upper level of system optimal results and other two provide less optimal solutions. Meanwhile, it has to be noticed that the solution with K-Means and K-Medoids Tabu algorithms do not necessarily to find a global optimum. The search technique might or might not find a good combination for all cases.

3. Due to the high computation cost, Tabu algorithm is more suitable to solve small scale problems rather than larger scale.

4. The process of decomposing problem size with K-Medoids and K-Means are based on different clustering principles. According to the performance of both algorithms, KMD-Tabu is more suitable for both case studies.

5. During the optimization process, cost value and waiting time value are related. When cost value is reducing by inserting requests, it may cause the violation of time window which increases the value of waiting time, contrarily, lower down the waiting time may increase travel cost.

Chapter 7

Conclusion and future work

This chapter provides a summary of the work and the conclusions that we have reached. Then, recommendations for the future work including aspects omitted in the case of study of the thesis.

7.1 Conclusion

This thesis contributes to the RACS routing problem by formulating the much necessary modelling aspect and by deriving algorithms into the solution procedure. First of all, we clarified the reservation based car sharing (RACS) system by analyzing in detail its components, operation strategies and operation objective, representing it as a multi-AVs provide pickup and delivery services with time window under a static circumstance, and its routing process denotes as a combinatorial optimization problem. In this system, demand, predicted travel time and range of service are known ahead of time. Based on the perfect information obtained, operator optimally plans the requests dispatching and routing based on the planning horizon periods.

In order to efficiently represent the routing optimization process, a AVS chain concept was proposed, which maximum includes 5 states of AVs and 5 links of vehicle route. With this concept, the whole routing processes has been decompose into sections, in which trip and relocation links are directly involved in the routing optimization. According to the objectives of minimizing the system travel cost and inconvenience for customers, AVS chain model was presented and applied into the problem modelling procedure. Moreover, mixed integer programming was applied for mathematically formulating this model with decision variables, objective functions and constraints.

Since our problem is a combinatorial optimization the property of NP hardness accompanies it. Consequently and in order to efficiently solve the problem and test the model effectiveness, meta-heuristic method of Tabu search has been utilized in the solution procedure. And with consideration of real life applications for large size demand, K-Means and K-Medoids partitioning clustering methodologies are combined with Tabu search to decompose the initial problem into smaller sub problems. And each of the sub problem could conduct a parallel calculation in order to reduce the computation cost, afterwards all results would be collected and generate the whole routing plan for the operation.

At last, the formulation and solution algorithms were explored in a series of test problem that showcased the effectiveness of Tabu, K-Mean Tabu and K-Medoids Tabu algorithms for the RACS routing problem. Then based on their performance of testing small scale and large scale demand problems including *Siouxfall* network and *NewYork* taxi network, the conclusion were found as:

1. Due to the nature of meta-heuristics method, the global optimal solution is not guaranteed.
2. For small scale problem, Tabu algorithm is more suitable with higher quality of solutions and acceptable computation cost;
3. For large scale problems, Tabu is no long applicable because of high computation cost, and between K-Means Tabu algorithm (KMN-Tabu) and K-Medoids Tabu (KMD-Tabu) algorithm, the latter can generate the best solution.
4. Tabu algorithm provides upper level of solution optimality. However, due to the high computation cost, its application is limited. The reason for lower quality solution of other two algorithm is due to the decomposition which limited the search space for optimization.
5. From the solution stability perspective, K-Medoids Tabu(KMD-Tabu) performs more robust than K-Means Tabu (KMN-Tabu) algorithm.
6. During the optimization process, cost value and waiting time value are related, when the diversification strategy starts to execute, cost value decreases with the insertion moves, nevertheless, may cause additional violation of time window, which increase the waiting time. Contrarily, same pattern happens.

7.2 Limitations

In the thesis, an autonomous car sharing routing model is established based on AVS chain concept with the objective to combined both system and customer concerns, and successfully proved its optimization effectiveness with real life problems. By applying three algorithms into the solution procedure, conclude the most appropriate solution method to each scale of problem. This model could be applied to single horizon planning or generate the initial routes plan for the RACS system operation. Nevertheless, the thesis poses many questions for which have not answer yet. On the modelling side, our model has certain limitations due to the assumption of perfect information circumstances, cost model(travel time) could be a critical issue since the traffic situation dynamic changes in real life, prediction of traffic flow is difficult. Meanwhile, system may need to plan with multiple horizons to form a longer operation plan. Moreover, some of the operation elements are not considered in the model, such as maintenance of vehicles, refueling and charging and accidents. On the solution side, the computation cost is still high and could not ensure the operation efficiency, more fast algorithms need to be created and utilized. Additionally, we have formulated the function of automatic adding extra vehicles when the system capacity is insufficient, however, due to the time limitation the test of this part is not include in this thesis.

7.3 Future work

There are many areas of this thesis can be improved in the future. Firstly, from the **modelling perspective** of RACS system routing optimization, multiple planning horizon and more complex problem setting should be examined in the future, such as other vehicle operating factors, cost of vehicle temporary parking, charging or refueling which could be more precisely reflect reality of operation processes. Secondly, since our model assumes perfect traffic information circumstances, future work could consider the stochastic optimization of real time traffic situations and congestion effects for travel time cost which more fits the reality, here efficient prediction methods need to be developed and apply to the model that makes it more robust. Nevertheless, our RACS system is operated without rejection of requests, which needs to increase the system capacity according to the demand, it may not be able to achieved in reality, therefore, rejection strategy could be developed based on system performance with fixed fleet size, where the selection of rejected requests process need to consider in optimization model.

Moreover, from the **solution perspective**, more algorithms can be applied and tested, compare them with different size of problems and find out the most appropriate ones according to computation time efficiency and solution quality.

Another important future work is extending the proposed model which accounts for more complex operation strategies and business model, since the RACS system is one of the subsystems in autonomous vehicle sharing, diverse **business models** can be developed based on this thesis by adding more complex operation strategies, such as combine ride sharing with RACS during the operation period, which could be utilized as temporal based. The system can separately apply both of them to peak hour and non-peak hour demand. So a switching model between the transition of two different types operation model need to be developed. Additionally, RACS static planning could only use for the pre-reserved requests before operation, afterwards, dynamically accept other requests and constantly adjusting routes with both static and dynamic optimization.

Bibliography

Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295 – 303, 2012. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2012.05.028. URL `http://www.sciencedirect.com/science/article/pii/S0377221712003864`.

Adriano Alessandrini, Andrea Campagna, Paolo Delle Site, Francesco Filippi, and Luca Persia. Automated vehicles and the rethinking of mobility and cities. *Transportation Research Procedia*, 5:145–160, 2015. ISSN 23521465. doi: 10.1016/j.trpro.2015.01.002. URL `http://dx.doi.org/10.1016/j.trpro.2015.01.002`.

Lukas Ambühl, Francesco Ciari, and Monica Menendez. What about space? A simulation based assessment of AVs impact on road space in urban areas. *16th Swiss Transport Research Conference*, (May), 2016. URL `http://www.strc.ch/2016/Ambuehl_EtAl.pdf`.

Prateek Bansal, Kara M. Kockelman, and Amit Singh. Assessing public opinions of and interest in new vehicle technologies: An Austin perspective. *Transportation Research Part C: Emerging Technologies*, 67:1–14, 2016. ISSN 0968090X. doi: 10.1016/j.trc.2016.01.019. URL `http://dx.doi.org/10.1016/j.trc.2016.01.019`.

Bbstabler. TransportationNetworks, 2018. URL `https://github.com/bstabler/TransportationNetworks`.

Slim Belhaiza. A Data Driven Hybrid Heuristic for the Dial-A-Ride Problem with Time Windows. 2017.

Michel Bierlaire. *Optimization: Principles and Algorithms*. EPFL Press, Lausanne, 2015.

Joschka Bischoff and Michal Maciejewski. Simulation of City-wide Replacement of Private Cars with Autonomous Taxis in Berlin. *Procedia Computer Science*, 83(Ant):237–244, 2016. ISSN 18770509. doi: 10.1016/j.procs.2016.04.121. URL `http://dx.doi.org/10.1016/j.procs.2016.04.121`.

Line Blander, Tommy Clausen, and David Pisinger. Synchronized dial-a-ride transportation of disabled passengers at airports. *European Journal of Operational*

Research, 225(1):106–117, 2013. ISSN 0377-2217. doi: 10.1016/j.ejor.2012.09.008. URL `http://dx.doi.org/10.1016/j.ejor.2012.09.008`.

Kris Braekers, An Caris, and Gerrit K. Janssens. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166–186, 2014. ISSN 01912615. doi: 10.1016/j.trb.2014.05.007. URL `http://dx.doi.org/10.1016/j.trb.2014.05.007`.

M. Chassaing, C. Duhamel, and P. Lacomme. An els-based approach with dynamic probabilities management in local search for the dial-a-ride problem. *Engineering Applications of Artificial Intelligence*, 48:119 – 133, 2016. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2015.10.002. URL `http://www.sciencedirect.com/science/article/pii/S0952197615002171`.

Boyd Cohen and Jan Kietzmann. Ride On! Mobility Business Models for the Sharing Economy. *Organization and Environment*, 27(3):279–296, 2014. ISSN 15527417. doi: 10.1177/1086026614546199.

Global Automotive Consumer and Baby Boomers. Car Sharing in Europe Business Models , National Variations and Upcoming Disruptions. *Monitor Deloitte*, (6), 2017.

Jean-François Cordeau. A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Operations Research*, 54(3):573–586, 2006. ISSN 0030-364X. doi: 10.1287/opre.1060.0283. URL `http://pubsonline.informs.org/doi/abs/10.1287/opre.1060.0283`.

Jean François Cordeau and Gilbert Laporte. The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms. *4or*, 1(2):89–101, 2003. ISSN 16142411. doi: 10.1007/s10288-002-0009-8.

Jean François Cordeau and Gilbert Laporte. The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007. ISSN 02545330. doi: 10.1007/s10479-007-0170-8.

Gonçalo Homem de Almeida Correia and Bart van Arem. Solving the User Optimum Privately Owned Automated Vehicles Assignment Problem (UO-POAVAP): A model to explore the impacts of self-driving vehicles on urban mobility. *Transportation Research Part B: Methodological*, 87:64–88, 2016. ISSN 01912615. doi: 10.1016/j.trb.2016.03.002.

Scott Corwin, Joe Vitale, Kelly Eamonn, and Elizabeth Cathles. The future of mobility - How transportation technology and social trends. page 32, 2015. URL `https://www2.deloitte.com/content/dam/Deloitte/de/Documents/manufacturing/DUP-{_}Future-of-mobility.pdf`.

Iman Dayarian, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem.

Transportation Research Part E: Logistics and Transportation Review, 95:95–123, 2016. ISSN 13665545. doi: 10.1016/j.tre.2016.09.004. URL http://dx.doi.org/10.1016/j.tre.2016.09.004.

Hussein Dia and Farid Javanshour. Autonomous Shared Mobility-On-Demand: Melbourne Pilot Simulation Study. *Transportation Research Procedia*, 22:285–296, 2017. ISSN 23521465. doi: 10.1016/j.trpro.2017.03.035. URL http://dx.doi.org/10.1016/j.trpro.2017.03.035.

Daniel J. Fagnant and Kara M. Kockelman. Preparing a nation for autonomous vehicles. *Transportation Research Part A journal*, 77(October):1–32, 2013. ISSN 0965-8564. doi: 10.1016/j.tra.2015.04.003. URL www.enotrans.org.

Daniel J. Fagnant and Kara M. Kockelman. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transportation Research Part C: Emerging Technologies*, 40:1–13, 2014. ISSN 0968090X. doi: 10.1016/j.trc.2013.12.001. URL http://dx.doi.org/10.1016/j.trc.2013.12.001.

Lei Feng, Elise Miller-Hooks, Paul Schonfeld, and Matthew Mohebbi. Optimizing ridesharing services for airport access. *Transportation Research Record: Journal of the Transportation Research Board*, 2467:157–167, 2014. doi: 10.3141/2467-17. URL https://doi.org/10.3141/2467-17.

Hillary Jeanette Ford. Shared Autonomous Taxis : Implementing an Efficient Alternative to Automobile Dependency. (June):104, 2012.

Thierry Garaix, Christian Artigues, Dominique Feillet, and Didier Josselin. Vehicle routing problems with alternative paths : An application to on-demand transportation. *European Journal of Operational Research*, 204(1):62–75, 2010. ISSN 0377-2217. doi: 10.1016/j.ejor.2009.10.002. URL http://dx.doi.org/10.1016/j.ejor.2009.10.002.

Veaceslav Ghilas, Emrah Demir, and Tom Van Woensel. A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. *Transportation Research Part B: Methodological*, 91:34–51, 2016. ISSN 01912615. doi: 10.1016/j.trb.2016.04.015. URL http://dx.doi.org/10.1016/j.trb.2016.04.015.

Fred Glover. Tabu search—part i. *ORSA Journal on Computing*, 1(3):190–206, 1989. doi: 10.1287/ijoc.1.3.190.

Wolfgang Gruel and Joseph M. Stanford. Assessing the Long-term Effects of Autonomous Vehicles: A Speculative Approach. *Transportation Research Procedia*, 13:18–29, 2016. ISSN 23521465. doi: 10.1016/j.trpro.2016.05.003. URL http://dx.doi.org/10.1016/j.trpro.2016.05.003.

Timo Gschwind and Stefan Irnich. Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. *Transportation Science*, 49(2):335–354, 2015. doi: 10.1287/trsc.2014.0531. URL `https://doi.org/10.1287/trsc.2014.0531`.

O. Guemri, B. Beldjilali, A. Bekrar, and G. Belalem. Two-stage heuristic algorithm for the large-scale capacitated location routing problem. *International Journal of Mathematical Modelling and Numerical Optimisation*, 7(1):97–119, 2016. doi: 10.1504/IJMMNO.2016.074375.

Lauri Häme. *Demand-Responsive Transport: Models and Algorithms.* 2013. ISBN 978-952-60-5163-5. URL `http://lib.tkk.fi/Diss/2013/isbn9789526051635/isbn9789526051635.pdf`.

Sin C. Ho, W. Y. Szeto, Yong Hong Kuo, Janny M.Y. Leung, Matthew Petering, and Terence W.H. Tou. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 0:1–27, 2018a. ISSN 01912615. doi: 10.1016/j.trb.2018.02.001. URL `https://doi.org/10.1016/j.trb.2018.02.001`.

Songguang Ho, Sarat Chandra Nagavarapu, Ramesh Ramasamy Pandi, and Justin Dauwels. An Improved Tabu Search Heuristic for Static Dial-A-Ride Problem. pages 3–8, 2018b. URL `http://arxiv.org/abs/1801.09547`.

Sebastian Hörl. Implementation of an autonomous taxi service in a multi-modal traffic simulation using MATSim. *Department of Energy and Environment Chalmers University of Technology*, (June):82, 2016. ISSN 09658564. doi: 10.13140/RG.2.1.2060.9523.

Sebastian Hörl. Agent-based simulation of autonomous taxi services with dynamic demand responses. *Procedia Computer Science*, 109:899–904, 2017. ISSN 18770509. doi: 10.1016/j.procs.2017.05.418. URL `http://dx.doi.org/10.1016/j.procs.2017.05.418`.

Hagai Ilani, Elad Shufan, Tal Grinshpoun, Aviad Belulu, and Alex Fainberg. A reduction approach to the two-campus transport problem. pages 587–599, 2014. doi: 10.1007/s10951-013-0348-7.

Siddhartha Jain and Pascal Van Hentenryck. Large neighborhood search for dial-a-ride problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6876 LNCS(m):400–413, 2011. ISSN 03029743. doi: 10.1007/978-3-642-23786-7{_}31.

K. B. Bergvinsdottir. The genetic algorithm for solving the dial-a-ride problemtle. 2004. URL `http://www2.imm.dtu.dk/pubdb/p.php?3229`.

Namwoo Kang, Fred M. Feinberg, and Panos Y. Papalambros. Autonomous Electric

Vehicle Sharing System Design. *Journal of Mechanical Design*, 139(1):011402, 2016. ISSN 1050-0472. doi: 10.1115/1.4034471. URL `http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4034471`.

Suleyman Karabuk. A nested decomposition approach for solving the paratransit vehicle scheduling problem. *Transportation Research Part B*, 43(4):448–465, 2009. ISSN 0191-2615. doi: 10.1016/j.trb.2008.08.002. URL `http://dx.doi.org/10.1016/j.trb.2008.08.002`.

Yannick Kergosien, Christophe Lenté, and Jean-Charles Billaut. A tabu search algorithm for solving a transportation problem of patients between care units, 2008.

Taehyeong Kim. Model and Algorithm for Solving Real Time Dial-a-Ride Problem. page 187, 2011.

Dominik Kirchler and Roberto Wolfler Calvo. A Granular Tabu Search algorithm for the Dial-a-Ride Problem. *Transportation Research Part B: Methodological*, 56:120–135, 2013. ISSN 01912615. doi: 10.1016/j.trb.2013.07.014.

Alain L Kornhauser. Uncongested Mobility for All New Jersey's Area-wide aTaxi System. *Operations Research and Financial Engineering Princeton University*, 2013. URL `http://orfe.princeton.edu/{~}alaink/NJ{_}aTaxiOrf467F12/ORF467F12aTaxiFinalReport{_}Draft.pdf`.

Rico Krueger, Taha H. Rashidi, and John M. Rose. Preferences for shared autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 69:343–355, 2016. ISSN 0968090X. doi: 10.1016/j.trc.2016.06.015. URL `http://dx.doi.org/10.1016/j.trc.2016.06.015`.

Albert Y. S. Lam, Yiu-Wing Leung, and Xiaowen Chu. Autonomous Vehicle Public Transportation System: Scheduling and Admission Control. 17(5):1210–1226, 2015. URL `http://arxiv.org/abs/1502.07242`.

Raphaël Lamotte, André de Palma, and Nikolas Geroliminis. On the use of reservation-based autonomous vehicles for demand management. *Transportation Research Part B: Methodological*, 99:205–227, 2017. ISSN 01912615. doi: 10.1016/j.trb.2017.01.003.

Gilbert Laporte, Jorge Riera-Ledesma, and Juan-José Salazar-González. A Branch-and-Cut Algorithm for the Undirected Traveling Purchaser Problem. *Operations Research*, 51(6):940–951, 2003. ISSN 0030-364X. doi: 10.1287/opre.51.6.940.24921. URL `http://pubsonline.informs.org/doi/abs/10.1287/opre.51.6.940.24921`.

Gilbert Laporte, Stefan Ropke, and Thibaut Vidal. Heuristics for the Vehicle Routing

Problem. *Vehicle Routing*, pages 87–116, 2014. doi: 10.1137/1.9781611973594.ch4. URL `http://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch4`.

Scott Le Vine, Alireza Zolfaghari, and John Polak. Carsharing: Evolution, Challenges and Opportunities. *ACEA - SAG 22 - Report*, (September):1–18, 2014.

Fabien Lehuédé, Renaud Masson, Sophie N Parragh, Olivier Péton, and Fabien Tricoire. A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society*, 65(7):983–1000, 2014. doi: 10.1057/jors.2013.17. URL `https://doi.org/10.1057/jors.2013.17`.

Michael W. Levin. Congestion-aware system optimal route choice for shared autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 82:229–247, 2017. ISSN 0968090X. doi: 10.1016/j.trc.2017.06.020. URL `http://dx.doi.org/10.1016/j.trc.2017.06.020`.

Michael W Levin, Tianxin Li, Stephen D Boyles, and Kara M. Kockelman. A general framework for modeling shared autonomous vehicles. *95th Annual Meeting of the Transportation Research Board*, (January):1–23, 2016. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2017.04.006.

Todd Litman. Autonomous Vehicle Implementation Predictions: Implications for Transport Planning. *Transportation Research Board Annual Meeting*, 42(2017):36–42, 2017. doi: 10.1613/jair.301.

Todd Litman. Autonomous Vehicle Implementation Predictions: Implications for Transport Planning. *Transportation Research Board Annual Meeting*, 42(2018):36–42, 2018.

Mengyang Liu, Zhixing Luo, and Andrew Lim. A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological*, 81:267–288, 2014. ISSN 01912615. doi: 10.1016/j.trb.2015.05.009. URL `http://dx.doi.org/10.1016/j.trb.2015.05.009`.

C. B. Lucasius, A. D. Dane, and G. Kateman. On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasiblity and comparison. *Analytica Chimica Acta*, 282(3):647–669, 1993. ISSN 00032670. doi: 10.1016/0003-2670(93)80130-D.

Ying Luo and Paul Schonfeld. A rejected-reinsertion heuristic for the static Dial-A-Ride Problem. *Transportation Research Part B: Methodological*, 41(7):736–755, 2007. ISSN 01912615. doi: 10.1016/j.trb.2007.02.003.

Jiaqi Ma, Xiaopeng Li, Fang Zhou, and Wei Hao. Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. *Transportation*

Research Part C: Emerging Technologies, 84:124–141, 2017. ISSN 0968090X. doi: 10.1016/j.trc.2017.08.022. URL `http://dx.doi.org/10.1016/j.trc.2017.08.022`.

Mohamed Amine Masmoudi, Kris Braekers, Malek Masmoudi, and Abdelaziz Dammak. A hybrid Genetic Algorithm for the Heterogeneous Dial-A-Ride Problem. *Computers and Operations Research*, 81:1–13, 2017. ISSN 03050548. doi: 10.1016/j.cor.2016.12.008.

Neda Masoud and R Jayakrishnan. Formulations for Optimal Shared Ownership and Use of Autonomous or Driverless Vehicles. *Transportation Research Board*, pages 1–17, 2016. URL `https://trid.trb.org/view/1393683`.

Neda Masoud and R. Jayakrishnan. A decomposition algorithm to solve the multi-hop Peer-to-Peer ride-matching problem. *Transportation Research Part B: Methodological*, 99:1–29, 2017. ISSN 01912615. doi: 10.1016/j.trb.2017.01.004.

Yves Molenbruch, Kris Braekers, and An Caris. Benefits of horizontal cooperation in dial-a-ride services. *Transportation Research Part E*, 107:97–119, 2017. ISSN 1366-5545. doi: 10.1016/j.tre.2017.09.001. URL `https://doi.org/10.1016/j.tre.2017.09.001`.

Santiago Muelas, Antonio Latorre, and José María Peña. A variable neighborhood search algorithm for the optimization of a dial-a-ride problem in a large city. *Expert Systems with Applications*, 40(14):5516–5531, 2013. ISSN 09574174. doi: 10.1016/j.eswa.2013.04.015. URL `http://dx.doi.org/10.1016/j.eswa.2013.04.015`.

Santiago Muelas, Antonio LaTorre, and José María Peña. A distributed VNS algorithm for optimizing dial-a-ride problems in large-scale scenarios. *Transportation Research Part C: Emerging Technologies*, 54:110–130, 2015. ISSN 0968090X. doi: 10.1016/j.trc.2015.02.024. URL `http://dx.doi.org/10.1016/j.trc.2015.02.024`.

NYCTLC. TAXI DATA, 2018. URL `http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`.

Julie Paquette, Jean François Cordeau, Gilbert Laporte, and Marta M.B. Pascoal. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transportation Research Part B: Methodological*, 52:1–16, 2013. ISSN 01912615. doi: 10.1016/j.trb.2013.02.007. URL `http://dx.doi.org/10.1016/j.trb.2013.02.007`.

Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. Variable neighborhood search for the dial-a-ride problem. *Computers and Operations Research*, 37(6):1129–1138, 2010. ISSN 03050548. doi: 10.1016/j.cor.2009.10.003. URL `http://dx.doi.org/10.1016/j.cor.2009.10.003`.

J. Pearl. Heuristics: Intelligent search strategies for computer problem solving. 10(12), 1 1984.

Mm Peden, R Scurfield, D Sleet, and D Mohan. World report on road traffic injury prevention. *Kenya*, 12(5):267–78, 2004.

Victor Pimenta, Alain Quilliot, Hélène Toussaint, and Daniele Vigo. Models and algorithms for reliability-oriented Dial-a-Ride with autonomous electric vehicles. *European Journal of Operational Research*, 257(2):601–613, 2017. ISSN 03772217. doi: 10.1016/j.ejor.2016.07.037.

Ming-yao Qi, Li-xin Miao, Le Zhang, and Hua-yu Xu. A new Tabu Search heuristic algorithm for the Vehicle Routing Problem with Time Windows. *2008 International Conference on Management Science and Engineering 15th Annual Conference Proceedings*, (70702003):1648–1653, 2008. doi: 10.1109/ICMSE.2008.4669126. URL `http://ieeexplore.ieee.org/document/4669126/`.

Yuan Qu and Jonathan F Bard. The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C*, 32:1–20, 2013. ISSN 0968-090X. doi: 10.1016/j.trc.2013.03.007. URL `http://dx.doi.org/10.1016/j.trc.2013.03.007`.

Pierre-Jean Rigole. Study of a Shared Autonomous Vehicles Based Mobility Solution in Stockholm. 2014. URL `http://www.diva-portal.org/smash/record.jsf;jsessionid=kS81CXqhOs0_dmrLKUFTDaNVPVCycHsxfNNBxBRF.diva2-search7-vm?pid=diva2:746893&dswid=4734`.

Ulrike Ritzinger, Jakob Puchinger, and Richard F. Hartl. Dynamic programming based metaheuristics for the dial-a-ride problem. *Annals of Operations Research*, 236(2):341–358, 2016. ISSN 15729338. doi: 10.1007/s10479-014-1605-7.

Schönberger. Scheduling constraints in dial-a-ride problems with transfers : a metaheuristic approach incorporating a cross-route scheduling procedure with postponement opportunities. pages 243–272, 2017. doi: 10.1007/s12469-016-0139-6.

Susan Shaheen, Nelson Chan, Apaar Bansal, and Adam Cohen. Shared Mobility a Sustainablity and Technology Workshop: Definition, Industry Development and Early Understanding. *University of California Berkeley Transportation Sustainability Research Center*, page 30, 2015.

Chung Wei Shen and Luca Quadrifoglio. Evaluating centralized versus decentralized zoning strategies for metropolitan ada paratransit services. *Journal of Transportation Engineering*, 139(5):524–532, 2013. ISSN 0733-947X. doi: 10.1061/(ASCE)TE.1943-5436.0000529.

Spieser, Kevin, Kyle Ballantyne, Treleaven, Rick Zhang, Daniel Morton, and Marco Pavone. Toward a Systematic Approach to the Design and Evaluation of Automated Mobility-on-Demand Systems : A Case Study in Singapore The MIT Faculty has made this article openly available . Please share Citation Accessed Citable Link Detailed Terms Toward a Syst. pages 0–16, 2014.

Stefan Røpke. The dial a ride problem (DARP). 3(February):1–25, 2005.

Adam Stocker and Susan Shaheen. Shared Automated Vehicles: Review of Business Models. *The International Transport Forum*, pages 1–29, 2017. URL `https://www.itf-oecd.org/sites/default/files/docs/shared-automated-vehicles-business-models.pdf`.

P. Toth and D. Vigo. *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2002. ISBN 9780898715798. URL `https://books.google.de/books?id=TeMgA5S74skC`.

Marjolein Veenstra, Marilène Cherkesly, Guy Desaulniers, and Gilbert Laporte. The pickup and delivery problem with time windows and handling operations. *Computers and Operations Research*, 77:127–140, 2017. ISSN 03050548. doi: 10.1016/j.cor.2016.07.014. URL `http://dx.doi.org/10.1016/j.cor.2016.07.014`.

José Viegas, Luis Martinez, Philippe Crist, and Sharon Masterson. Shared Mobility. Inovation for Liveable Cities. *International Transport Forum's Corporate Partnership Board*, pages 1–56, 2016. doi: http://dx.doi.org/10.1787/5jlwvz8bd4mx-en.

Warren E. Walker and Vincent A.W.J. Marchau. Dynamic adaptive policymaking for the sustainable city: The case of automated taxis. *International Journal of Transportation Science and Technology*, 6(1):1–12, 2017. ISSN 20460430. doi: 10.1016/j.ijtst.2017.03.004. URL `http://linkinghub.elsevier.com/retrieve/pii/S2046043016300156`.

Hao Wang, Ruey Long Cheu, and Der Horng Lee. Intelligent taxi dispatch system for advance reservations. *Journal of Public Transportation*, 17(3):115–128, 2014. ISSN 1077291X. doi: 10.5038/2375-0901.17.3.8. URL `http://www.scopus.com/inward/record.url?eid=2-s2.0-84908040654&partnerID=tZOtx3y1`.

Wenwen Zhang, Subhrajit Guhathakurta, Jinqi Fang, and Ge Zhang. Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach. *Sustainable Cities and Society*, 19:34–45, 2015. ISSN 22106707. doi: 10.1016/j.scs.2015.07.006.

Issam Zidi, Kamel Zidi, Khaled Mesghouni, and Khaled Ghedira. *A multi-agent system based on the multi-objective simulated annealing algorithm for the static dial a ride problem*, volume 18. IFAC, 2011. ISBN 9783902661937. doi: 10.3182/20110828-6-IT-1002.02639. URL `http://dx.doi.org/10.3182/20110828-6-IT-1002.02639`.