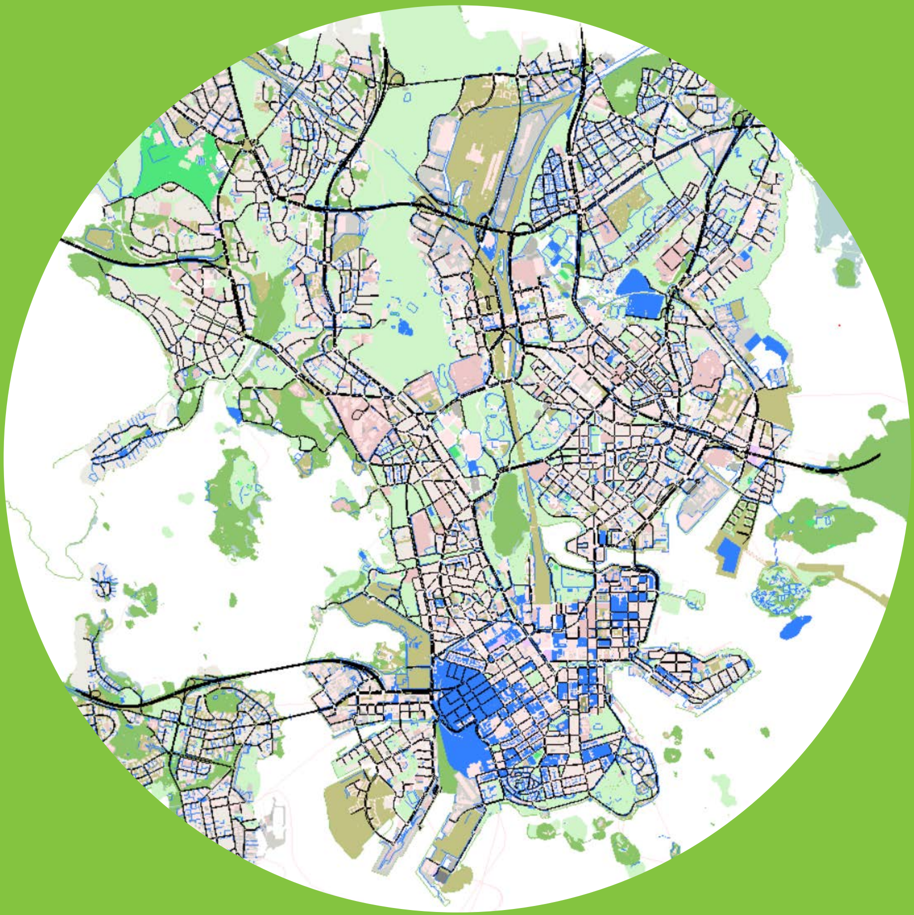
Department of Information and Communications Engineering

Capacity Planning for Vehicular Fog Computing

Wencan Mao



Aalto University

DOCTORAL THESES

Aalto University publication series
DOCTORAL THESES 172/2023

Capacity Planning for Vehicular Fog Computing

Wencan Mao

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall AS1, Maarintie 8 on 3 November 2023 at 12:00.

Aalto University
School of Electrical Engineering
Department of Information and Communications Engineering
Mobile Cloud Computing

Supervising professors
Yu Xiao, Aalto University, Finland
Antti Ylä-Jääski, Aalto University, Finland

Thesis advisor
Özgür Umut Akgül, Nokia Corporation. Finland

Preliminary examiners
Weisong Shi, University of Delaware, USA
Eirini Eleni Tsiropoulou, University of New Mexico, USA

Opponent
Schahram Dustdar, TU Wien, Austria

Aalto University publication series
DOCTORAL THESES 172/2023

ISBN 978-952-64-1481-2 (printed)
ISBN 978-952-64-1482-9 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-64-1482-9

Unigrafia Oy
Helsinki 2023

Finland

NORDIC SWAN ECOLABEL
Printed matter
4041-0619

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Wencan Mao

Name of the doctoral thesis
Capacity Planning for Vehicular Fog Computing

Publisher School of Electrical Engineering

Unit Department of Information and Communications Engineering

Series Aalto University publication series DOCTORAL THESES 172/2023

Field of research Networking Technology

Manuscript submitted 25 September 2023 **Date of the defence** 3 November 2023

Permission for public defence granted (date) 7 September 2023 **Language** English

□ **Monograph** ☒ **Article thesis** □ **Essay thesis**

Abstract

The strict latency constraints of emerging vehicular applications make it unfeasible to forward sensing data from vehicles to the cloud for processing. Fog computing shortens the network latency by moving computation close to the location where the data is generated. Vehicular fog computing (VFC) proposes to complement stationary fog nodes co-located with cellular base stations (i.e., CFNs) with mobile ones carried by vehicles (i.e., VFNs) in a cost-efficient way. Previous works on VFC have mainly focused on optimizing the assignments of computing tasks among available fog nodes. However, capacity planning, which decides where and how much computing resources to deploy, remains an open and challenging issue. The complexity of this problem results from the spatio-temporal dynamics of vehicular traffic, the uncertainty in the computational demand, and the trade-off between the quality of service (QoS) and cost expenditure.

This dissertation focuses on capacity planning for VFC. The objective of capacity planning is to maximize the techno-economic performance of VFC in terms of profit and QoS. To address the spatial-temporal dynamics of vehicular traffic, this dissertation presents a capacity planning solution for VFC that jointly decide the location and number of CFNs together with the route and schedule of VFNs carried by buses. Such a long-term planning solution is supposed to be updated seasonally according to the traffic pattern and bus timetables. To address the uncertainty in the computational resource demand, this dissertation presents two capacity planning solutions for VFC that dynamically schedule the routes of VFNs carried by taxis in an on-demand manner. Such a short-term planning solution is supposed to be updated within minutes or even seconds. To evaluate the techno-economic performance of our capacity planning solutions, an open-source simulator was developed that takes real-world data as inputs and simulates the VFC scenarios in urban environments. The results of this dissertation can contribute to the development of edge and fog computing, the Internet of Vehicles (IoV), and intelligent transportation systems (ITS).

Keywords

ISBN (printed) 978-952-64-1481-2 **ISBN (pdf)** 978-952-64-1482-9

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2023

Pages 134 **urn** http://urn.fi/URN:ISBN:978-952-64-1482-9

Preface

I would like to express my special gratitude to my supervisor, Prof. Yu Xiao, who offered me the opportunity to join her research team at Aalto University. Her guidance in the broad field of scientific research, as well as her invaluable advice, support, and encouragement, have greatly assisted me in completing this dissertation.

I greatly appreciate the help of my co-supervisor, Prof. Antti Ylä-Jääski, who provided me with invaluable support during my doctoral research and gave me excellent guidance during the paper publication and teaching.

I warmly thank my advisor Dr. Ozgur Akgul, co-authors Dr. Byungjin Cho, Dr. Abbas Mehrabi, Jiaming Yin, Yushan Liu, Prof. Yang Chen, and Prof. Weixiong Rao. This dissertation was made possible by your great contributions, insightful discussions, and constructive feedback. It was my great pleasure to work with you.

I extend my gratitude to my colleagues for contributing to a pleasant working environment. Particularly, thanks to Xuebing Li, Truong-an Pham, Petr Byvshev, Aziza Zhanabatyrova, Clayton Frederick Souza Leite, Emmi Pouta, Henry Mauranen, Tim Moesgen, Esa Vikberg, Zeyu Yang, Jiayun Zhang, Hui Ruan, Min Gao, Dr. Giancarlo Pastor, Dr. Ramyah Gowrishankar, and Dr. Chao Zhu.

Special thanks to Prof. Weisong Shi from the University of Delaware and Prof. Eirini Eleni Tsiropoulou from the University of New Mexico for pre-reviewing my dissertation. Your insightful comments and valuable feedback have helped me improve the quality of this dissertation that meets the high standards of the scientific community.

My sincerest appreciation is directed to Prof. Schahram Dustdar from the TU Wien, who kindly accepted to act as my opponent in the public defense of my dissertation.

I appreciate the financial support of the School of Science and School of Electrical Engineering.

Over the last four years, I have learned that pursuing a doctoral degree is a long and arduous journey, especially for me with a background in other disciplines. It made me realize that the strongest support and love from my family are the keys to success. I send my warmest thanks to my loving family, especially my grandfather Songjin Li and my mother Bei Li, who build the warm harbor at home. I send my love to my husband Taiquan Liu and my three lovely cats. It was their endless love, patience, and encouragement that supported me during the toughest times and inspired me to achieve the end. Therefore, I dedicate this dissertation to them.

Espoo, September 25, 2023,

Wencan Mao

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Wencan Mao, Ozgur Umut Akgul, Abbas Mehrabi, Byungjin Cho, Yu Xiao, and Antti Ylä-Jääski. Data-Driven Capacity Planning for Vehicular Fog Computing. Accepted for publication in *IEEE Internet of Things Journal*, Volume: 9, Issue: 15, Pages: 13179 - 13194, August 2022.

II Ozgur Umut Akgul, Wencan Mao, Byungjin Cho, and Yu Xiao. VFogSim: A Data-driven Platform for Simulating Vehicular Fog Computing Environment. Accepted for publication in *IEEE Systems Journal*, Volume: 17, Issue: 3, Pages: 5002 - 5013, September 2023.

III Wencan Mao, Ozgur Umut Akgul, Byungjin Cho, Yu Xiao, and Antti Ylä-Jääski. On-demand Vehicular Fog Computing for Beyond 5G Networks. Accepted for publication in *IEEE Transactions on Vehicular Technology*, 1-17 pages, June 2023.

IV Wencan Mao, Jiaming Yin, Yushan Liu, Byungjin Cho, Yang Chen, Weixiong Rao, and Yu Xiao. Multi-agent Reinforcement Learning-based Capacity Planning for On-demand Vehicular Fog Computing. Submitted to *pre-review*, June 2023.

Author's Contribution

Publication I: "Data-Driven Capacity Planning for Vehicular Fog Computing"

The author of this dissertation was the lead architect of this project. She collected and analyzed the real-world datasets, designed the algorithm that optimizes the deployment of cellular fog nodes as well as vehicular fog nodes with fixed routes and timetables, and evaluated the proposed capacity planning framework via simulation.

Publication II: "VFogSim: A Data-driven Platform for Simulating Vehicular Fog Computing Environment"

The author of this dissertation was one of the key designers of this platform. She designed the task generation module in VFogSim with the first author, generated the input data including the vehicular traffic and communication channel parameters, and implemented the real-world measurements of 5G network parameters with the third author to validate the platform.

Publication III: "On-demand Vehicular Fog Computing for Beyond 5G Networks"

The author of this dissertation was the lead architect of this project. She collected and analyzed the real-world datasets, designed the algorithm that optimizes the deployment of vehicular fog nodes with flexible routes and timetables, and evaluated the proposed capacity planning framework via simulation.

Publication IV: "Multi-agent Reinforcement Learning-based Capacity Planning for On-demand Vehicular Fog Computing"

The author of this dissertation was the lead architect of this project. She collected and analyzed the real-world datasets, collaborated with the second author to design the algorithm that dynamically schedules the routes of vehicular fog nodes, and collaborated with the third author to evaluate the proposed capacity planning framework via simulation.

Abbreviations

3GPP 3^{rd} Generation Partnership Project

5G 5^{th} Generation of Mobile Telephony

AI Artificial Intelligence

AV Autonomous Vehicle

CFN Cellular Fog Node

CNN Convolutional Neural Network

CPU Central Processing Unit

C-V2X Cellular Vehicle-to-Everything

DQN Deep Q-Network

DSRC Dedicated Short-Range Communications

FL Federated Learning

GCN Graph Convolutional Network

gNB 3GPP-compliant implementation of the 5G-NR base station

GPS Global Positioning System

GPU Graphics Processing Unit

GTFS General Transit Feed Specification

HFP High-Frequency Positioning

HSL Helsinki Regional Transport Authority

ILP Integer Linear Programming

IoT Internet of Things

IoV Internet of Vehicles

ITS Intelligent Transportation Systems

LSTM Long Short-Term Memory

LTE Long-Term Evolution

MARL Multi-Agent Reinforcement Learning

MDP Markov Decision Process

MILP Mixed-Integer Linear Programming

NR New Radio

OBU On-Board Unit

ODVFC On-Demand Vehicular Fog Computing

QoS Quality of Service

RL Reinforcement Learning

RSU Roadside Unit

SARIMA Seasonal Autoregressive Integrated Moving Average

SINR Signal-to-Interference-plus-Noise Ratio

SLA Service Level Agreement

SUMO Simulation of Urban Mobility

TMS Traffic Measurement System

TraCI Traffic Control Interface

TTI Transmission Time Interval

UAV Unmanned Aerial Vehicle

V2N Vehicle-to-Network

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

VFC Vehicular Fog Computing

VFN Vehicular Fog Node

VRP Vehicle Routing Problem

VRPTW Vehicle Routing Problem with Time Windows

WLAN Wireless Local Area Network

XML Extensible Markup Language

1. Introduction

According to the Global Forecast report, the global market size of autonomous vehicles (AVs) is projected to grow from 20.3 million units in 2021 to 62.4 million units by 2030 [74]. The emerging vehicular applications, such as cooperative intersection crossing [14] and lane change scheduling [62], are usually compute-intensive and latency-sensitive. Due to space, weight, and cost constraints, most current vehicles may not have sufficient onboard computing capacity to handle such applications. If all the data is forwarded to the cloud for processing, the vehicles may fail to meet the low-latency requirements of such applications, due to the long and unstable round trip time. Fog and edge computing share the idea of moving computational resources closer to where the data is generated to reduce network latency [104]. In this dissertation, we refer to such a computing paradigm as fog computing.

In early works of fog computing, computing nodes were usually co-located with stationary infrastructure, such as cellular base stations [118], roadside units (RSUs) [6], or smart grids and traffic lights [101]. We refer to these stationary fog nodes as cellular fog nodes (CFNs) in this dissertation. One of the most significant concerns associated with the deployment of CFNs is that these nodes cannot be moved after deployment. Such a constraint raises issues in satisfying the vehicular computational demand, due to the spatio-temporal variation in vehicular traffic and the resource consumption of vehicular applications [131]. Some computing nodes may be severely swamped by overwhelming demand in particular regions (e.g., with dense road networks) at specific periods of time (e.g., during peak hours). On the other hand, adding more capacity requires additional investment. If the capacity is deployed based on the peak demand, this static provisioning would result in wasting resources at other times.

As an alternative approach, vehicular fog computing (VFC) has been proposed in [112] as a new computing paradigm where fog nodes are also installed on moving vehicles (e.g., buses and taxis). We refer to these mobile fog nodes as vehicular fog nodes (VFNs) in this dissertation. The key idea of VFC is to complement the stationary CFNs with moving VFNs

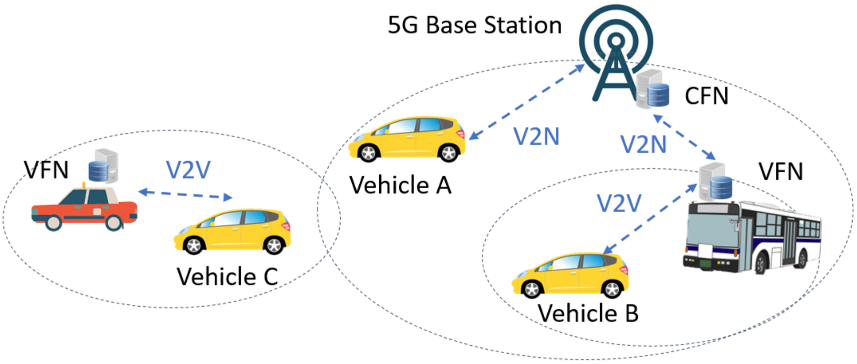


Figure 1.1. An exemplary scenario of VFC, where a CFN co-located with a cellular base station can serve *Vehicle A* within a one-hop vehicle-to-network (V2N) communication range, a VFN carried by a bus is scheduled according to its route and timetable to serve *Vehicle B* within a one-hop vehicle-to-vehicle (V2V) communication range, and a VFN carried by a taxis is routed to the destination in advance to serve *Vehicle C* within a one-hop V2V communication range.

to reduce costs while fulfilling the quality of service (QoS) requirements of vehicular applications. Fig. 1.1 illustrates an exemplary scenario of VFC, where there are dual-direction communications between the client vehicles (i.e., vehicles generating computational tasks) and fog nodes, and computation can be offloaded from client vehicles to either a CFN or a VFN within a one-hop communication range. We limit it to one-hop communication to guarantee low latency [35, 38]. Some VFNs have fixed routes and timetables, while others do not, depending on whether the vehicles carrying the fog nodes need to follow fixed routes and timetables. In urban environments where vehicular traffic follows certain spatio-temporal distributions, VFC lowers the overall installation and operational costs and reduces the occurrence of service migration between fog nodes by moving VFNs along with the client vehicles [112].

Previous works on VFC focus mainly on *task allocation*, which determines the assignments of computing tasks among available fog nodes. For example, Zhu et al. [132] proposed a joint optimization solution to assign the tasks generated by client vehicles to different CFNs and VFNs under service latency, quality loss, and fog capacity constraints, with the assumption that the capacity of each CFN or VFN is predefined. *Capacity planning*, which determines the location and capacity of the fog nodes to deploy, remains an open and challenging issue. This dissertation considers two types of VFNs, including the types with fixed routes and timetables, such as bus-carried VFNs, and the types with flexible routes and timetables, such as taxi-carried VFNs. The capacity planning for VFC considers the location and capacity of CFNs as well as the routes and time schedules of VFNs. The complexity of this problem results from the spatio-temporal dynamics of vehicular traffic, uncertainty in computational demand, and trade-offs between QoS and costs.

The goal of capacity planning for VFC is to maximize the techno-economic performance in terms of QoS and profit. QoS is evaluated by performance metrics, including end-to-end latency (i.e. the total latency from generating tasks to receiving results) [112] and service rate of computational tasks (i.e., the percentage of tasks served within the latency requirement). Profit equals revenue (i.e., cumulative price) minus costs and penalties. Prices and penalties are defined in the service-level agreement (SLA) [52]. Costs include installation costs (e.g., hardware and site costs) and operating costs (e.g., travel, rental, and maintenance costs). The research questions, scope, methodology, and contributions of this dissertation are elaborated below.

1.1 Research Questions and Scope

In this dissertation, the main research question is: How can the deployment plans of CFNs and VFNs be optimized in order to maximize the techno-economic performance of VFC? The computational demand generated by the client vehicles depends on the distribution of vehicular traffic and the resource usage of different vehicular applications on each vehicle.

According to the analysis of real-world data obtained from HERE [20], we found that the vehicular traffic in Helsinki, for example, follows certain spatio-temporal patterns. Specifically, we observe two peak hours (i.e., morning and afternoon) during the weekdays versus one peak hour (i.e., around noon) during the weekends; meanwhile, the traffic density in the city downtown is usually higher than that in the suburb. In addition to this, there are uncertainties in vehicular traffic, in terms of variance in traffic flow distribution and density over time. For example, traffic may be affected by weather conditions (e.g., summer versus winter), working style (e.g., onsite, remote, or hybrid mode), or occasional events (e.g., music festivals and football matches).

Considering that there are no real-world statistics or references about future usages of different vehicular applications, we assume that the arrival of computing tasks (i.e., how many computing tasks would be generated by each vehicle) and the selection of vehicular application (i.e., which applications each vehicle would prefer to use) follow a uniform distribution, and the consumption rate of computational resources depends on the type of application in question and its corresponding QoS requirement.

Due to the uncertainty in vehicular traffic, there could be occasional changes in the computational demand generated by vehicles. To timely adapt the capacity plan to the changes, the model needs to be sufficiently lightweight. Therefore, the time complexity of the capacity planning solutions is another feature this dissertation focuses on. To solve the problems step-by-step, the research question has been divided into three sub-questions.

- **RQ1** How to fulfill the computational demand generated by the client vehicles with minimum installation and operational costs of CFNs and VFNs, assuming that the vehicular traffic follows certain spatio-temporal patterns?

  RQ1 concerns long-term capacity planning. To answer this question, it is necessary to analyze the spatio-temporal variation in the demand and supply of VFC resources, design theoretical models and algorithms for capacity planning, and evaluate how the spatio-temporal variation will impact the techno-economic performance of the capacity plan.

- **RQ2** How to adapt the capacity plan to the occasional change in vehicular traffic, with the aim of maximizing the techno-economic performance?

  RQ2 concerns short-term capacity planning. To answer this question, one approach is to proactively predict future demand and update the capacity plan that maximizes the techno-economic performance in the upcoming moment. Another approach is to learn a capacity planning policy (i.e., a mapping from the current environment observation to a probability distribution of the actions to be taken) that aims to maximize the long-term techno-economic performance, taking the uncertainty of demand into account.

- **RQ3** How to balance the time complexity of the capacity planning solution and the optimality of the capacity plan? In other words, how to ensure that the capacity plan can be updated within the time requirement from the capacity planning perspective while achieving high techno-economic performance?

The broad scope of this dissertation inevitably necessitates omitting certain topics and challenges. First, this dissertation assumes that the CFNs and VFNs have homogeneous hardware specifications and provide identical services to the users. Nevertheless, by calculating the minimum capacity needed in different locations, we can easily translate the number of fog nodes with identical capacity into the number of resource units on a single fog node; thus, the heterogeneity problem can be solved by applying simple capacity conversion. Second, we assume that the arrival of computing tasks and the selection of vehicular applications follow a uniform distribution. Despite this, the capacity planning solutions we propose can also be used with unevenly distributed computing tasks, since the demand estimation is used as input for the capacity planning model. Third, this dissertation assumes one-hop communication between each fog node and the vehicles within its communication range, leaving multi-hop communications out of scope. Furthermore, we do not consider cloud-fog hybrid infrastructure where computational tasks can be executed collaboratively between fog and cloud [37]. Finally, the security and privacy issue [60, 40, 58] is another topic that is beyond the scope of this dissertation.

1.2 Methodology

This dissertation follows a data-driven methodology [11] as shown in Fig. 1.2. In such a methodology, strategic decisions are based on the analysis and interpretation of data. First, we observe the spatio-temporal patterns and the uncertainty in the vehicular traffic data, which motivates the research questions described in Section 1.1. To solve the research questions, we obtain vehicular traffic data and application profiles from various sources, based on which we model the computational demand. Meanwhile, to plan for the deployment solutions of the resources, we collect data in terms of road networks, cellular base stations, and commercial fleets. Based on the analysis and interpretation of the above data, capacity planning decisions are made. Finally, we evaluate the decisions through simulations, using the collected data as inputs.

The data-driven methodology consists of the following steps:

- **Define the Problem**: We define the capacity planning problem (i.e., research questions detailed in Section 1.1).

- **Acquire the Data**: We identify the data required to address the problem. More specifically, we collect real-world data, including the road maps from HERE map [21] and OpenStreetMap [23], cellular base station maps from OpenCellID [22] and CellMapper [18], vehicular traffic datasets from HERE Traffic API [20], traffic measurement system (TMS) data [19], Helsinki regional transport authority (HSL) open data [25], and vehicular application profiles (i.e., resource consumption of each vehicular application) from benchmark testing.

- **Process the Data**: The collected road maps, cellular base station maps, and vehicular traffic data are processed using various spatial-temporal analysis methods, such as traffic flow theory [32], graph theory [51], k-means clustering [87], Gaussian process regression [95], and seasonal autoregressive integrated moving average (SARIMA) [53].

- **Model the Problem**: We formulate the capacity planning problem using integer linear programming (ILP) [54] in Publications I and III. More specifically, Publication I aims to minimize the installation and

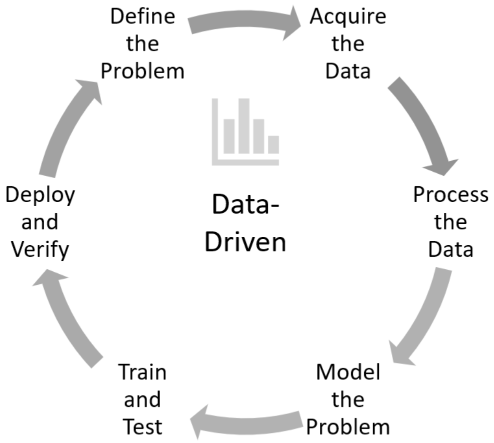


Figure 1.2. The data-driven methodology in use [11].

operational costs of CFNs and VFNs, taking QoS requirements, resource capacity, and communication range as constraints. Publication III uses resource capacity and communication range as constraints and aims at maximizing the techno-economic performance of VFNs. We model the capacity planning problem as a Markov decision process (MDP) [105] in Publication IV, where the VFNs are regarded as agents who interact with the VFC environment to maximize the techno-economic performance.

- **Train and Test**: We train and test the model with samples from the collected data. In Publications I and III, we adjust the formulation of each ILP module for obtaining the cost or profit-optimal capacity plan. In Publication IV, we deploy different reinforcement learning architectures for training the VFN routing policies. We repeat the above process for several iterations to fine-tune the model.

- **Deploy and Verify**: We use the simulator developed in Publication II to evaluate the proposed capacity planning solution and compare it with baseline algorithms. In Publications I and III, the models output the capacity plans of different types of fog nodes, QoS measurements, and cost or profit estimation. In Publication IV, we apply the trained VFN routing policies to the VFC simulations to obtain the QoS measurements and profit estimation. We analyze the impacts on techno-economic performance from various factors, such as resource availability, economic models, and spatio-temporal variations of traffic flow. We also investigate the time complexity of the proposed capacity planning solution.

1.3 Contributions

This section summarizes four publications that address the proposed research questions. To address **RQ1**, Publication I proposes a long-term capacity planning framework. The framework estimates the spatio-temporal demand based on regression from real-world traffic data as well as application profiles obtained through benchmark testing. It outputs a cost-optimal deployment plan of CFNs and VFNs using a heuristic algorithm and ILP. The result of this work shows the potential of complementing CFNs with VFNs to fulfill the dynamic computational demand with lower costs.

To address **RQ2**, Publication III proposes an on-demand VFC (ODVFC) scenario, where the VFNs are carried by taxis and routed to the places where demand emerges. Such an ODVFC scenario can quickly adapt the routes of VFNs to temporary changes in demand. We use SARIMA to predict the real-time vehicular traffic flow and formulate the capacity planning problem as ILP. To address **RQ3**, a two-phase capacity planning model is adopted to plan the city-scale routing strategies of VFNs and the proportion of computing tasks to be served on a daily basis, followed by a region-scale routing of VFNs and task allocation from client vehicles to VFNs within each region. The two-phase capacity planning enables parallel planning at the regional level, which is sufficiently lightweight to be updated frequently.

Reinforcement learning (RL) is a computational method in which an agent takes actions that will result in state transitions in the environment, and receives observations of new states and rewards from the environment. RL can be used to solve very complex problems and address the curse of dimensionality of traditional optimization problems (i.e., the time complexity greatly increases with the size of variables) [105], thus being suitable for capacity planning for VFC. Multi-agent reinforcement learning (MARL) refers to the RL problem where multiple agents interact in the same environment to achieve a common goal [86]. In the case of ODVFC, each VFN is regarded as an agent.

Therefore, as an alternative solution to address **RQ2**, Publication IV proposes a MARL approach based on the actor-critic [66] to achieve dynamic routing of VFNs in ODVFC. The proposed solution can adapt the routes of VFNs to the fast-changing demand and dynamic traveling time. To further address **RQ3**, the considered actor-critic method achieves low time complexity by using decentralized policies while enabling collaboration among agents by using centralized training.

To evaluate the proposed capacity planning solutions, Publication II develops an open-source simulator called VFogSim, which extends the existing vehicular simulation platforms [36, 100, 121] to support realistic network simulation, mobility of fog nodes, customizable computational and network schedulers, and techno-economic analysis. It allows real-

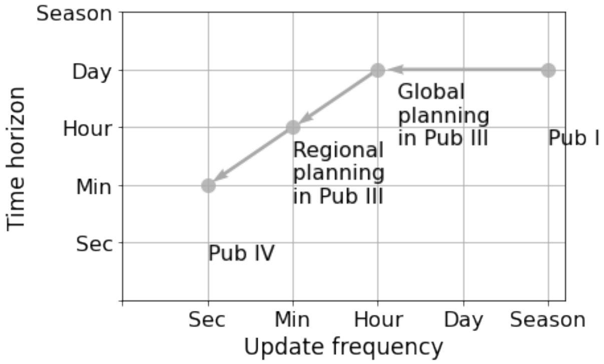


Figure 1.3. Relationship among Publications I, III, and IV, where the x-axis represents the update frequency (i.e., how often the capacity plan is updated), and the y-axis represents the time horizon (i.e., the considered time range for capacity planning).

world data as input for simulating the supply and demand of VFC in urban areas. Publication IV extends VFogSim into Ray-SUMO-VFogSim++, which supports new features, such as 5G new radio (NR) V2X and MARL environment.

Overall, this research moves towards reducing the time granularity of capacity planning, as shown in Fig. 1.3. Since the time horizon and update frequency in Publication II are flexible, the proposed simulator is suitable for evaluating all the capacity planning solutions.

Publication I creates a capacity plan that defines the locations and capacity of each CFN as well as schedules the daily trips of bus-carried VFNs. The plan is valid for a season, during which there is no significant change in the bus schedules or traffic flow distributions. One season typically lasts for one or several months. Publication III considers two-phase planning. During the global planning phase, the region-to-region routes of each VFN during the day are created. The routes are updated on an hourly basis. At the regional planning phase, the region-wise segment-to-segment routes of VFNs in the following hour are created. These routes are updated every several minutes. Finally, Publication IV considers the continuous update (i.e., every second) of the destination of each VFN during each interval of regional planning. The decreasing time granularity also leads to stricter time requirements for capacity planning. Therefore, the time complexity of the model is analyzed, especially in Publications III and IV.

In practice, the proposed methods can be applied sequentially from Publications I, III, and IV to deploy and refine the capacity plan. Alternatively, the regional planning model in Publication III could be replaced with the MARL-based approach in Publication IV and increase the time horizon to an hour. This will reduce the time complexity of the solution and improve scalability, at the expense of slightly lowering techno-economic performance.

1.4 Structure

The remainder of this dissertation is organized as follows. Chapter 2 introduces the concept and related technology of VFC, followed by previous works on demand prediction, task allocation, and capacity planning. It also reviews the research on vehicle routing and VFC-related simulation platforms. Chapter 3 summarizes the main contributions of the dissertation. Chapter 5 concludes the work. The original publications are presented after Chapter 5.

2. Background

This chapter presents the background knowledge needed to understand the domain covered in this dissertation. First, Section 2.1 introduces the concept and related technology of VFC. Then, Sections 2.2, 2.3, and 2.4 provide an overview of the latest works on demand prediction, task allocation, and capacity planning, respectively. Considering that capacity planning for VFC focuses on scheduling the routes of VFNs according to demand distribution, Section 2.5 reviews the works on vehicle routing in the operational research domain. Section 2.6 reviews the simulation platforms relevant to VFC. Finally, Section 2.7 summarizes the main points of the chapter.

2.1 Vehicular Fog Computing

Vehicular Fog Computing (VFC) refers to the computing paradigm that deploys computing capacity on vehicles. VFC facilitates a wide range of vehicular services, such as smart traffic control, road safety improvement, video crowd-sourcing, and entertainment services [44, 112, 83]. In extreme cases, each vehicle will be equipped with a VFN. However, due to high installation and operational costs, it is more realistic to assume that VFNs will be deployed on some vehicles to serve other vehicles in the vicinity. So far, there is no uniform definition or regulations on where to deploy VFNs. In previous research, some [42] proposed to deploy VFNs on parked vehicles, while others [112] proposed to deploy VFNs on moving ground vehicles or unmanned aerial vehicles (UAVs).

By moving computational and communication resources closer to the client vehicles, VFC achieves better communication efficiency and addresses the limitations of traditional cloud-based vehicular networks in terms of latency, location awareness, and real-time response [44]. Furthermore, by exploiting the mobility of VFNs, VFC can provide cost-effective on-demand fog computing services [112]. Note that this dissertation does not consider the scenario of offloading computing tasks to the cloud.

Despite this, we propose to run capacity planning algorithms that are computational-expensive but not latency-sensitive (e.g., long-term planning) in the cloud.

From the network perspective, a VFN is equipped with an on-board unit (OBU) with one or multiple network interfaces. Previous work [112] has assumed that each VFN connects to the Internet through the cellular network, and it is also configured as a wireless local area network (WLAN) access point and/or a gateway for nearby mobile devices to connect to the Internet [112]. Currently, dedicated short-range communications (DSRC) and cellular vehicle-to-everything (C-V2X) are the most widely used radio access technologies for vehicular communication.

DSRC is an IEEE 802.11p-based wireless communication technology that enables highly secure, high-speed direct communication between vehicles and the surrounding infrastructure [78]. DSRC uses an orthogonal frequency division multiplexing-based physical layer with a channel bandwidth of 10 MHz [80]. IEEE 802.11bd further extends IEEE 802.11p and aims to support vehicular applications characterized by high-reliability and low-latency requirements [80, 78].

C-V2X is the technology developed within the 3rd Generation Partnership Project (3GPP) [2] and designed to operate in vehicle-to-vehicle (V2V) and vehicle-to-network (V2N) modes [80]. C-V2X uses the widely distributed cellular infrastructure and defines additional transmission modes that allow direct V2V communication using side-link channels [80]. Different from DSRC, which only supports broadcast, C-V2X enables the exchange of messages between vehicles, pedestrians, and wayside traffic control devices, such as traffic signals. Furthermore, 5G NR V2X [1], which is an upgraded version of Long Term Evolution (LTE)-V2X, is featured in ultra-high reliability, ultra-low latency, high throughput, flexible mobility, and energy efficiency [5]. Therefore, in this dissertation, we assume that 5G NR V2X is used for implementing communications between vehicles and with the network infrastructure. More specifically, as shown in Fig. 1.1, the client vehicles and VFNs send their real-time information, such as location, speed, and computing task or resource capacity, to the connected cellular base stations periodically through 5G NR V2N. The CFNs co-located with the cellular base stations coordinate the task allocation from the client vehicles to VFNs and CFNs. The communications and task offloading between client vehicles and VFNs are implemented through 5G NR V2V.

2.2 Demand Prediction Problem

Capacity planning starts by predicting the demand for computing resources at different times of day in different locations. The spatio-temporal distribution of demand depends on the distribution of vehicular traffic (since vehicles are the ones generating the computational workload) and the resource consumption rate of the vehicular applications in use. The vehicular application profile describes the usage pattern of the hardware resource, including central processing unit (CPU), graphics processing unit (GPU) [85, 84], and memory [132].

According to traffic flow theory [32], a road network is viewed as a graph where nodes represent road segments and edges represent intersections, or vice versa. Traffic flow describes both macroscopic and microscopic behaviors of vehicular traffic. The former can be measured with traffic volume (i.e., the number of vehicles passing a road segment over a period of time), traffic density (i.e., the number of vehicles located on a road segment at a specific time), or average speed (i.e., the average speed of the vehicles passing a road segment over a period of time). The latter refers to the mobility (e.g., moving trajectory and speed) of individual vehicles.

From the capacity planning perspective, macroscopic traffic flow prediction is enough for estimating the number of computing resources needed in different areas. The traffic flow prediction problem aims to predict the short-term macroscopic traffic flow (e.g., in the upcoming hour) using the time series of historical data as inputs. Classic time series analysis techniques, such as SARIMA [53] and time series decomposition [45], have been widely used to solve the problem with high prediction accuracy.

In recent years, machine learning has also been applied to the traffic flow prediction problem. To learn the spatial correlations and features of the road network, one approach is to use k-nearest neighbor [68], which selects the most related neighbors to learn the patterns of the testing road segment. Another method is to apply convolutional neural network (CNN) [125, 29], which is a widely-used tool to learn spatial features and dependencies. Considering the graph structure of the road network, graph convolutional network (GCN) [125, 103, 88] has also been used to learn the spatial features as a generalization of CNNs. To learn the temporal features in the traffic flow, long short-term memory (LSTM) is currently a commonly used network. Apart from discrete data samples, LSTM can also process well in the time sequence, thus suitable for traffic flow prediction. In addition to the above structures, embedding (i.e., a low dimensional vector to represent certain features) is also used to learn various features from real-world data to improve the traffic flow prediction accuracy, including weather conditions [125, 114], traveling time matrix [27], congestion propagation patterns [29], and traffic incidents [114]. Machine learning-based methods improve accuracy compared to classic methods. However, the training of the neural network is usually data-intensive and time-consuming.

Regarding vehicular application profiles, it is impossible to profile all possible vehicular applications or to know in advance which applications each vehicle would be using. In practice, a common approach is to choose exemplary applications and benchmark the resource consumption of these applications. The results would be used as references for estimating the resource consumption rate of each vehicle. From the capacity planning perspective, it is necessary to know how the achievable QoS, e.g., latency, changes with the amount of capacity available. Therefore, we created vehicular application profiles as a model of resource consumption versus QoS requirement. Various regression methods can be used for modeling, such as linear or non-linear least squares regression [132].

2.3 Task Allocation Problem

As described in Chapter 1, task allocation focuses on how to assign computing tasks to different computing nodes, with the assumption that the capacity of each computing node is pre-defined. Resource allocation is necessary to evaluate the techno-economic performance of the capacity plan. Therefore, this section reviews the research related to task allocation in VFC, where the objective is to maximize the QoS, such as minimizing latency.

Considering the strict time requirement of making task allocation decisions, heuristic methods are often used [116], including particle swarm optimization [132, 41], matching theory [126, 115], and game theory [50]. For example, Zhu et al. [132] designed a dynamic task assignment framework for VFC in which binary particle swarm optimization was used to jointly optimize service latency and quality loss. Hou et al. [41] used fault-tolerant particle swarm optimization for computational offloading and task assignment among CFNs and VFNs to maximize the reliability with latency constraints. Zhou et al. [126, 128, 127] proposed a two-stage VFC framework with a contract theory-based resource management mechanism and a matching learning-based task offloading mechanism. Peng et al. [89] presented an auction-based resource allocation mechanism to improve the efficiency and authenticity of VFC. Overall, Heuristics reduce time complexity compared to optimization such as ILP. However, they suffer from sub-optimality.

As an alternative solution, machine learning has been applied to task assignments in VFC. For example, Singh et al. [98] proposed a resource allocation technique for software-defined network-enabled VFC using collaborative machine learning, where the data owned by the client vehicles are trained in a distributed and parallel manner. Apart from this, decentralized RL [56, 15, 109, 113] has also been utilized. In this case, the problem is formulated as an MDP, where each fog node is an agent. The agent takes action at each time step and interacts with the VFC environment to learn its own task allocation policy that maximizes the QoS. For example, Zhao et al. [124] designed a contract-based resource allocation mechanism based on distributed deep RL and queuing theory in order to reduce system complexity and avoid decision collision. Cho et al. [17] presented an online task offloading algorithm that aims to minimize offloading service costs in terms of latency and energy while considering time-varying resource supply and demand. Zhu et al. [129] proposed a delay- and resolution-aware task offloading strategy for visual-based assisted driving based on a partially observable Markov decision process (MDP) and solved the problem using a Monte Carlo method. They [130] further proposed a context-aware task assignment scheme to jointly optimize the quality of information and processing latency for vehicle-based visual crowdsourcing using deep Q-learning (DQN). Liu et al. [63] formulated resource allocation in VFC as a semi-MDP and used DQN to maximize long-run utility. Although the time complexity of decentralized learning is low, it suffers from non-stationary issues, since the actions that an agent takes also affect the states of the other agents.

To address the non-stationary issue while maintaining a low time complexity, actor-critic [117], and its extended algorithm, deep deterministic policy gradient [28], have been applied to the MARL environment. In this case, the VFNs can learn their task allocation policies with the help of a centralized agent. For example, Yang et al. [117] proposed a task scheduling algorithm based on actor-critic to realize the coordination of multiple UAV tasks in a UAV swarm. Dai et al. [28] jointly optimized the task offloading, bandwidth, and computational resource allocation decisions in VFC using deep deterministic policy gradient. Inspired by these works, we applied the actor-critic method to short-term capacity planning (i.e., ODVFC) in Publication IV.

2.4 Capacity Planning Problem

The capacity planning problem dates back to the context of cloud computing, where service-level agreements (SLAs) are commonly used. The SLA defines negotiable parameters, such as desired QoS, prices, and penalties, which provide inputs for calculating the most profitable capacity plan that avoids or minimizes the violations of the agreement [52]. For example, Kouki et al. proposed an SLA-driven capacity planning framework for cloud applications, where they used a queuing-based model to predict the cloud service performance and find the optimal configuration using a utility function [52]. Similarly, Ranaldo et al. proposed a capacity-driven utility model for SLA negotiation of cloud services, taking into account the desired QoS and expected resource availability, costs, and penalties [94]. Fog computing moves the computational capacity closer to the edge to shorten the latency, but it suffers from unconventional issues such as the limited number and coverage of computing resources in distributed locations.

Capacity planning in edge/fog computing environments is usually formulated as an optimization problem with different objectives, inputs, outputs, and constraints. One approach to model and solve capacity planning problems is to use classical methods (e.g., the knapsack algorithm). For example, Noreikis et al. [85] built a knapsack-based capacity planning model for edge computing, aiming at meeting QoS requirements while minimizing the number of required fog nodes. They [84] further applied the queuing theory to capacity planning for real-time compute-intensive applications. However, these methods cannot be directly applied to VFC because they do not consider the mobility of vehicles, including vehicles generating computational demands and vehicles carrying computational resources. Boualouache et al. [7] developed a software-defined network paradigm and a stochastic model to estimate the number of fog nodes to be deployed in a given area. However, they only considered stationary resource provision.

Alternatively, heuristics can be used for capacity planning. For example, Chiu et al. [16] deployed a heuristic algorithm to simultaneously decide the number of fog nodes with appropriate communication resource allocation and computational task allocation. Zhang et al. [122] proposed a framework that employed weighted sums and evolutionary algorithms to optimize the trade-off between capital expenditure and network latency. Haider et al. [37] utilized weighted sum, hierarchical, and trade-off methods to simultaneously determine the optimal location, capacity, number of fog nodes, and connections between fog nodes and the cloud to minimize latency in the network and the traffic to the cloud. While providing fast results, heuristic solutions are often sub-optimal.

Another solution for capacity planning is to use ILP or mixed integer linear programming (MILP). Stypsanelli et al. [102] proposed an optimal capacity planning solution for fog computing infrastructure under probabilistic latency guarantees, aiming to save energy and operational costs. Hussein et al. [47] aimed to find the optimal location and capacity of fog nodes to minimize overall network latency and energy consumption. Premsankar et al. [91] aimed to minimize the deployment cost of edge devices by jointly meeting the network coverage requirement and computational demand of vehicular applications in smart cities. By using ILP or MILP, one can guarantee an optimal solution. However, the execution time can be very long due to high computational complexity, especially when the number of fog nodes and client vehicles is high.

Similar to moving ground vehicles, capacity planning for UAVs has also been studied. Luo et al. [67] presented a deep RL-based solution to jointly make optimal computation offloading decisions and flying direction selection for multi-UAV collaborative target search. Sun et al. [103] utilized UAVs for video analytics tasks, and they proposed a flexible and lightweight genetic algorithm to determine the task assignment solution as well as the dynamic position of the UAVs to reduce the execution time and energy consumption. Shen et al. [97] proposed a MARL-based multi-UAV cooperative search algorithm that is trained on a digital twin of the area in order to maximize the search rate and coverage rate. Hu et al. [43] designed a two-layered optimization algorithm to maximize the total computation bits for VFC, where the outer layer optimizes the resource allocation from the client vehicles to CFNs and VFNs, while the inner layer optimizes the trajectory scheduling of VFNs carried by UAVs. Unlike UAVs, which travel freely in a continuous space, ground vehicles have to travel on road networks and follow traffic regulations, which adds a layer of complexity to the capacity planning problem in this dissertation. Besides, none of the above works considered the economic model or techno-economic analysis.

2.5 Vehicle Routing Problem

The vehicle routing problem (VRP) in the operational research domain aims to minimize the travel cost of a vehicle that is supposed to complete a set of tasks by optimizing its route. On this basis, VRP with time windows (VRPTW) considers the scenario with multiple vehicles and the dynamic demand that changes over time. Capacity planning for VFC considers scheduling the routes of VFNs according to the predicted distribution of demand for computing resources, similar to the VRP and VRPTW problems. However, our problem differs from theirs in terms of objectives and constraints. Our goal is to maximize the techno-economic performance of VFC under the constraints of computing capacity, delay requirements, and communication range.

The VRP has been applied to vehicular crowdsensing. Ding et al. [30] presented a MARL framework for crowd-sensing in urban vehicles, where vehicles that collect the sensing data make distributed routing decisions and collaboratively optimize system-wide revenue, sensing coverage, and sensing quality. Likewise, Du et al. [31] developed a distributed deep-learning method for trajectory prediction to maximize sensing coverage and accuracy in VFC. Different from them, we consider computation offloading from client vehicles to VFNs and the routing of VFNs.

Previously, VRP has been studied in the context of VFC. Yuan et al. [120] considered the stationary fog nodes and proposed a DQN-based MARL algorithm to jointly decide service migration and routing decisions of client vehicles. They aimed to meet the service delay requirements with minimal migration cost and travel time. They focused on the routing of client vehicles, whereas we focused on the routing of VFNs which provide computing services for client vehicles.

Although ILP has been widely used to solve VRP and VRPTW, it suffers from high time complexity since the problems are NP-hard. Therefore, heuristics such as genetic algorithms [90], particle swarm optimization [90], and modified A* algorithm with Haversine and Vincenty formulas [4] have been used to reduce execution time, at the cost of lower optimality.

Recently, single-agent deep RL was deployed in [88, 82, 119] to solve the VRP, where the vehicle is the agent who learns its routing policy independently. They viewed the road network as a graph where each node is a road segment, and an instance is a set of nodes that make up a route. Peng et al. [88] regarded the partial solutions of instances and the features of each node in the graph structure as states and used a dynamic attention model with RL to find routing solutions. Nazari et al. [82] considered both dynamic states (i.e. customer's location) and static states (i.e. customer's demand), and they applied actor-critic to train routing policy. Yin et al. [119] proposed a node quality predictor and graph decomposition-based method to find the shortest paths on large dynamic graphs. However, in the case of jointly planning the routes for multiple vehicles, the routing solution would become sub-optimal due to the non-stationary issue.

The centralized learning-based approach was applied in [123, 57] to solve the multi-vehicle VRP, where they used a single agent to train a global routing policy. Zhang et al. [123] leveraged a multi-agent attention model combined with policy gradients to solve the problem, taking into account both the agent state (i.e., vehicle's location and remaining capacity) and the environment state (i.e., unvisited customers and depot locations). Lee et al. [57] focused on heterogeneous VRP (i.e., where vehicles have different capacities) and used policy gradients to solve the problem. Both of the above-mentioned works used the attention model to reduce the time complexity since the centralized methods suffer from exploding states and action spaces.

2.6 Simulation Platforms

To plan the deployment of VFC in the real world, we need to consider the dynamic demand and supply of computing resources, as well as the trade-offs between QoS and potential installation and operating costs. Given the complexity and economic pressures of real-world measurements, simulation becomes a better option at an early research stage to validate capacity planning solutions in various urban environments. Therefore, this section reviews the existing VFC-related simulation platforms.

Current edge/fog computing simulators are mainly built on top of existing cloud computing simulators or network simulators. Three well-known edge computing simulators are IFogSim [36], IoTSim [121] and EdgeCloudSim [100], which are all built upon CloudSim [12]. IFogSim models a fog environment, where fog nodes follow a hierarchical arrangement from the sensor to the cloud, and measures the impact of resource management policies on latency, network congestion, energy consumption, and cost [36]. IoTSim supports simulating the Internet of Things (IoT) big data processing using the MapReduce model [121]. However, it simplifies network models and does not consider communication channel properties such as signal-to-interference-plus-noise ratio (SINR). EdgeCloudSim integrates multiple modules into an edge computing system, including the core simulation module (i.e., responsible for loading and running edge computing scenarios from configuration files), network module, load generator, user mobility module, and edge coordinator [100]. Although user mobility is considered, it does not support the mobility of fog nodes.

FogNetSim++ [92] is an edge computing simulator based on OMNet++ [107], which focuses on simulating the network characteristics of distributed edge computing devices. It supports user-defined mobility models and fog node scheduling algorithms [92]. However, it does not support the estimation of network metrics, such as throughput, due to the lack of a physical layer protocol. Similarly, EmuFog [75] and Fogbed [26] are two edge computing simulators based on the network simulator Mininet [55] and its extended version MaxiNet [110]. EmuFog enables users to design network typologies with embedded fog nodes and runs Docker-based applications [77] on these connected nodes via a simulated network [75]. Fogbed supports dynamically adding, connecting, and removing virtual nodes via Docker containers and includes real-world protocols and services [26]. Although the above simulators support the evaluation of cost and latency, they do not support other advanced features, such as customizable schedulers, pricing policies, and inter-service prioritization.

FogTorch [9] is a simulation tool that enables QoS-aware deployment of IoT applications to fog infrastructure. FogTorchII [10] extends FogTorch, leverages Monte Carlo simulations to account for QoS variations, and classifies deployments based on QoS guarantees and fog resource consumption. However, neither of them integrates the mobility model of user vehicles, nor considers the mobility of fog nodes. To simulate vehicular networks, Veins [99] combines OMNeT++ [107] with the mobile simulator SUMO [129] to implement IEEE 802.11p. It can also be used with SimuLTE [108] and Simu5G [81], which provide detailed models of LTE-based and 5G NR-based V2X, respectively. However, it does not contain computational schedulers, service-specific network schedulers, or economic models.

The aforementioned simulators could not be directly used in RL since there is a lack of interactions between the simulation environments and agents. In recent years, new simulation platforms have been introduced that combine vehicular traffic simulation with RL libraries through online interaction. For example, the Flow project [111] integrates the traffic simulator SUMO [65] with Ray RLlib [79], which could be used to evaluate acceleration and routing strategies of vehicles as well as control strategies of traffic signals. However, it could not be applied to VFC because it lacks vehicle network simulation. Veins-Gym [96] is developed based on Veins [99] and Open AI Gym [8]. It supports a scenario where a vehicle communicates with others via a vehicular ad-hoc network and chooses among different communication modules according to the reception probability and transmission cost [96]. However, Veins-Gym does not support MARL scenarios, 5G network simulations, or task offloading between vehicles.

2.7 Summary

This chapter has reviewed the state-of-the-art research related to capacity planning for VFC. The principle findings are summarized below.

VFC has been a new computing paradigm that draws increasing research attention, where vehicles are used as carriers of fog nodes to provide computing and communication resources on demand. Moreover, the rapidly developing vehicular communication technologies, especially in 5G, have facilitated the potential of VFC in real-world implementations. Task allocation and capacity planning are usually coupled together when managing the demand and supply of computational resources. While task allocation in VFC has been a heated topic over recent years, capacity planning for VFC remains to be an open and challenging issue.

Capacity planning for VFC needs to jointly consider the spatiotemporally varying demand and supply of computational resources. In terms of predicting the computational demand, both classic and machine learning-based methods have been widely studied in the literature for traffic flow prediction, and vehicular application profiles can be used to model the resource consumption patterns of various applications.

The supply of computational resources, from the capacity planning perspective, is usually formulated as an optimization problem with different objectives and constraints. From the method perspective, ILP- and MILP-based methods can guarantee optimal solutions but have high time complexity; on the other hand, heuristic and RL-based methods have lower time complexity but are sub-optimal. Existing works on capacity planning for cloud, edge, and fog computing utilized various optimization methods, but none of them have considered the mobility of fog nodes.

The VRP in the operational research domain aims to optimize the routes of vehicles that are supposed to complete a set of tasks in order to minimize travel costs. Similar to the VRP, capacity planning for VFC considers scheduling the routes of VFNs according to the predicted distribution of demand for computing resources. However, we aim to maximize the techno-economic performance of VFC, taking into account the constraints of computing capacity, communication range, and QoS requirements.

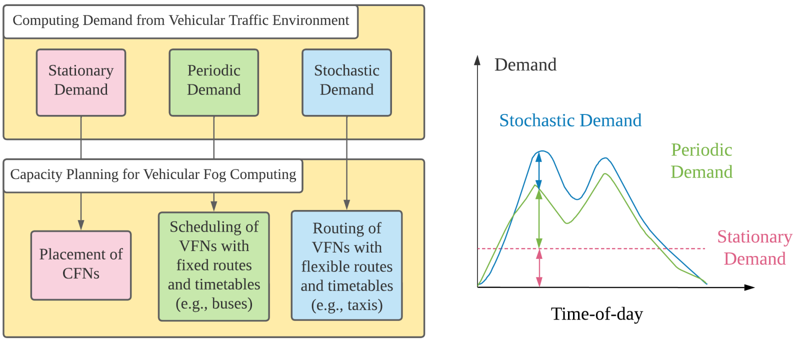
Finally, considering the early phase of VFC, the real-world experiment would be expensive and time-consuming. Since existing simulators lack key features in VFC, such as realistic vehicular network simulation, computational and network schedulers, and economic models, a new simulation platform is necessary to validate the proposed capacity planning solutions under urban environments.

3. Capacity Planning for Vehicular Fog Computing

This chapter summarizes the contributions of each publication and shows how they address the research questions introduced in Chapter 1. Section 3.1 presents an overview of our capacity planning solutions. Section 3.2 describes the impacts of vehicular mobility on capacity planning in the VFC environment. Section 3.3 describes the ILP-based long-term capacity planning solution that addresses **RQ1**. Sections 3.4 and 3.5 explain the ILP-based and MARL-based capacity planning solutions for ODVFC, respectively, which target **RQ2** and **RQ3**. Section 3.6 illustrates the simulation platforms that we have developed for evaluating the solutions explained in the earlier sections.

3.1 Overall Framework

Fig. 3.1 overviews the dissertation work. According to the study on real-world traffic dataset [20], we found that the computational demand generated by vehicular traffic is accumulated from three parts, including stationary, periodic, and stochastic demands [71].



Computing Demand from Vehicular Traffic Environment
Stationary Demand
Periodic Demand
Stochastic Demand
Capacity Planning for Vehicular Fog Computing
Placement of CFNs
Scheduling of VFNs with fixed routes and timetables (e.g., buses)
Routing of VFNs with flexible routes and timetables (e.g., taxis)
Demand
Stochastic Demand
Periodic Demand
Stationary Demand
Time-of-day

Figure 3.1. An overview of capacity planning for VFC to fulfill the stationary, periodic, and stochastic computational demands [71].

The stationary demand, as shown in pink in Fig. 3.1, is defined as the level of demand that is lower than the actual demand during most time of the day. We envision fulfilling this part of demand with CFNs, which can provide constant capacity at all times. Apart from this, the periodic demand, as shown in green in Fig. 3.1, follows certain spatio-temporal patterns that repeat daily on weekdays and weekly on weekends. We envision fulfilling this part of the demand by scheduling the VFNs carried by buses, which have fixed routes and timetables that are designed to follow traffic patterns (e.g., more frequent during peak hours). We learned the stationary and periodic demands from regressions on daily traffic flows, and the solution to meet them (**RQ1**) is addressed in Publication I.

The pattern of vehicular traffic could change due to seasonal variations (e.g., summer versus winter) or the occasional events (e.g., music festivals and football matches). For seasonal variations, we can update the long-term plan every season, but it is computationally heavy to update it every day or hour. For occasional events, it is not worth changing the long-term capacity plan because the changes are usually non-repetitive. To calculate the stochastic demand caused by these changes, as shown in blue in Fig. 3.1, we first estimate the overall demand by traffic flow prediction and then exclude the regular demand from the overall demand estimation. We envision deploying VFNs on vehicles with flexible routes and schedules (e.g., taxis), and routing them when and to where that demand emerges. The solutions to meet the stochastic demand (**RQ2**) are addressed in Publications III and IV, and the time complexity analyses of the proposed solutions (**RQ3**) are also presented. Finally, the simulator proposed in Publication II and extended in Publication IV is used to evaluate the proposed capacity planning solutions.

3.2 Impacts of Vehicular Mobility

The vehicles involved in VFC include the client vehicles and VFNs. As mentioned in Section 2.2, the mobility of vehicles can be studied from two perspectives: the macroscopic and microscopic flows. The microscopic flow refers to the moving trajectory and speed of individual vehicles. The macroscopic flow refers to the aggregation of individual vehicles and the interactions among them.

The distribution of client vehicles follows the macroscopic flow. The macroscopic flow can be measured with traffic volume, traffic density, or average speed. In this dissertation, we use traffic density to represent the number of client vehicles in each region, which is derived from traffic volume data [19] and speed data [20] according to the traffic flow theory [32]. In addition, Gaussian process regression is used for long-term (e.g., seasonal) macroscopic flow modeling, while SARIMA is used for short-term (e.g., hourly) macroscopic flow prediction.

Based on the distribution of client vehicles, we decide on the mobility of VFNs as our capacity plan. The mobility of VFNs depends on the routes and timetables of the vehicles carrying them. If we use buses to carry the VFNs, the routes and timetables of bus trips are determined and can be extracted by open source data [25]. Therefore, we decide on the bus trips that will carry the VFNs and the turnaround sequence of the selected bus trips as our capacity plan. If we use taxis to carry the VFNs, the routes and timetables of the VFNs are flexible. Therefore, we update the origins and destinations of the VFNs regularly as our capacity plan.

Apart from the macroscopic flow, the microscopic flow is also necessary to understand the implications of mobility on the capacity plan. The microscopic model of bus-carried VFNs depends on the trajectory of buses, which is collected from the onboard GPS data [25]. We use SUMO [65] to simulate the microscopic models of client vehicles and taxis-carried VFNs, where each vehicle is associated with an origin and destination (OD) matrix and their trajectories are calculated according to the shortest path on the road network. The OD matrix of client vehicles is set based on their commuting schedule or daily activity, while the capacity plan provides the OD matrix of VFNs. The speeds of the vehicles are updated according to the car-following model, affected by road regulations (e.g., speed limit) and traffic infrastructures (e.g., traffic lights). When the location of a vehicle changes, its network attributes, such as connected cellular base station, SINR, and throughput values, also vary, based on which we decide the resource allocation plan and evaluate the techno-economic performance.

3.3 ILP-based Long-term Capacity Planning

To address **RQ1**, we propose a data-driven capacity planning framework in Publication I, which takes real-world data as inputs and outputs a cost-optimal deployment plan of CFNs and VFNs using a heuristic algorithm and ILP. It is built on the assumption that CFNs are co-located with cellular base stations located in different areas of the city while VFNs are carried by buses with fixed routes and timetables. We use three types of data as inputs. The vehicular traffic data include the speed samples from HERE traffic API [20] and traffic volume samples from the Finnish traffic measurement system (TMS) [19]. The application profiles describe the CPU and GPU consumption of each vehicular application versus latency requirement. The bus mobility data include real-time bus locations and bus timetables, obtained from HSL high-frequency positioning (HFP) API [25] and HSL general transit feed specification (GTFS) [25], respectively.

Our framework determines the number and types of fog nodes to be deployed in different regions (e.g., downtown and suburban areas) to

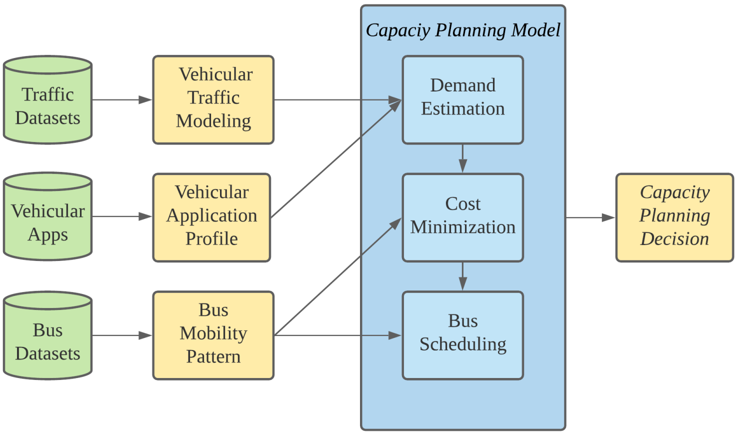


Figure 3.2. Flowchart of the data-driven capacity planning framework in Publication I [73].

meet computational resource requirements at different times of the day, under the constraints of resource capacity, communication range, and QoS requirements. Since not all buses must be equipped with VFNs to fulfill the computational demand, the framework decides which bus lines and shifts will be served by VFN-equipped buses.

Fig. 3.2 outlines the data-driven capacity planning framework. We implement the capacity planning model in three steps: demand estimation, cost minimization, and bus scheduling. Vehicular traffic data and application profiles are used for demand forecasting. Bus mobility data (including routes and timetables) is used for cost minimization and bus scheduling.

We first estimated the computational resource demand generated by client vehicles, which depends on the spatio-temporal distribution of vehicular traffic and the resource consumption of vehicular applications. Based on real-world traffic datasets, we calculated traffic density using the traffic flow theory [32]. We then used k-means clustering [87] to divide the considered city into clusters and used Gaussian process regression [95] to model the daily traffic flow within each cluster. Meanwhile, we selected exemplary vehicular applications, containerized them into Docker [77] images, and analyzed their resource usage under different latency constraints. The output of the demand estimation module defines the minimum computing capacity (expressed as the number of fog nodes with a fixed unit capacity) required by each cluster to meet the QoS requirements.

Our second step was to determine a cost-optimal deployment plan based on the estimated demand and potential supply. Based on real-world bus schedules, we mapped bus trips that pass through each cluster using a spatio-temporal availability matrix (i.e., a binary value is calculated for each bus trip in each cluster at each timeslot, where 1 indicates available and 0 indicates unavailable). Here, a bus trip defines the route traveled and when the trip starts. The same trip is usually repeated daily on weekdays and weekly on weekends. The same bus trip may be served by different buses on different days.

Our cost minimization module generated CFN deployment plans (i.e., the number and locations of the CFNs) and bus trip selection (i.e., which bus trips should provide the VFC service) that minimized the operational costs of CFNs and VFNs. In this module, we assumed that all buses in the study area carry VFNs. However, this could lead to an oversupply of VFNs.

Our final step was to run the bus scheduling module to determine the smallest subset of buses to cover the selected bus trips for VFN deployment. Bus trips that belong to the same bus line and have sufficient turnaround times were linked together to ensure that the same bus can cover different trips. In this way, we minimized the installation costs of VFNs.

We used the simulator described in Section 3.6 to simulate the VFC scenarios in Helsinki downtown versus suburban areas, on weekdays versus weekends. The experimental results proved the cost-efficiency potential of VFC over deploying only CFNs in the long term. The deployment of VFNs saves operational costs at the expense of additional installation costs. Compared to deploying VFNs on all buses in the study area, the bus scheduling module prevents unnecessary installation costs by choosing a subset of buses to carry VFNs. We also evaluated the impacts of traffic patterns on the capacity plans and the potential cost savings. We found that high traffic density and significant hourly variation would lead to dense deployment of VFNs and create more savings in operational costs in the long term.

3.4 ILP-based Short-term Capacity Planning

The method proposed in Publication I focuses on long-term capacity planning and is not suitable for short-term capacity planning due to high time complexity. In addition, the fixed routes and schedules of buses cannot accommodate the uncertainty in computational demand.

Therefore, to address **RQ2**, we propose an ODVFC scenario in Publication III, where the VFNs carried by vehicles with flexible routes and schedules (e.g., taxis) are routed at the time and to the places where occasional demand emerges. In ODVFC, the capacity plan needs to be updated frequently, which requires the capacity planning solution to be lightweight enough to complete routing calculations within minutes or even seconds.

To ensure the availability of VFNs in a given region when delay-critical services are received, we assumed that VFNs are routed to the region before demand generation. In order to realize this assumption, it is necessary to be able to accurately predict the traffic flow and corresponding computing

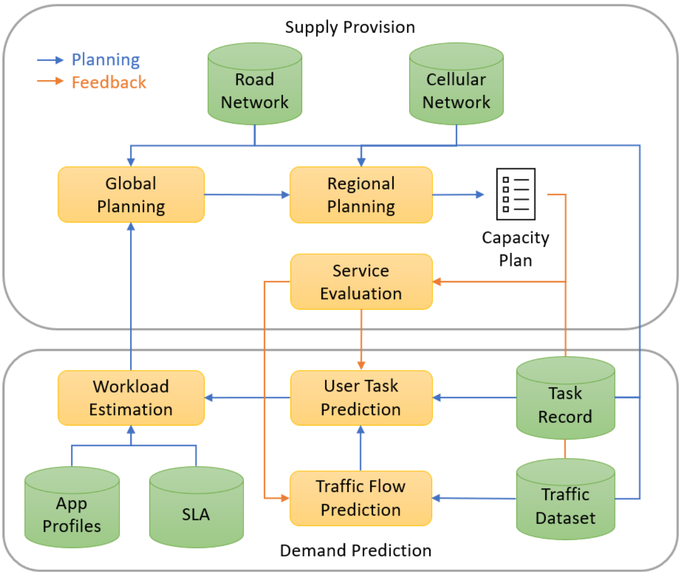


Figure 3.3. An overview of the capacity planning framework in Publication III [72].

resource demand in the next time slot (e.g., hour).

Fig. 3.3 presents the data-driven framework to implement capacity planning in ODVFC, where the demand and supply are modeled separately. The inputs for demand prediction are the traffic dataset, task record, application profiles, and SLA.

In Publication I, we assumed the spatio-temporal distribution of demand remains the same, whereas in Publication III, we considered the temporary changes in the distribution of demand. Therefore, we applied SARIMA [53] to the traffic dataset for traffic flow prediction, and used the task record, which describes the task arrival rate and user selections of vehicular applications, to compute the task arrival. Combined with vehicular application profiles describing CPU and GPU consumption patterns for each task, workloads can be estimated. In other words, we estimated how much CPU and GPU capacity is needed to meet the latency requirements of the applications running on the vehicles in each region.

Moreover, in Publication I, we assumed that all the computing tasks would be served by CFNs and VFNs. However, in ODVFC, due to the limited number and capacity of VFNs carried by taxis and high variation in the stochastic demand, not all computing tasks could be served, especially when the increase in demand is significant. Planning to serve all the demands leads to resource scarcity as well as SLA violations. Therefore, we chose the most profitable proportion of demand to be served according to the prices and penalties defined in the SLA.

To address **RQ3**, a two-phase capacity planning model is deployed. During global planning, we determined the global routing strategy that aims

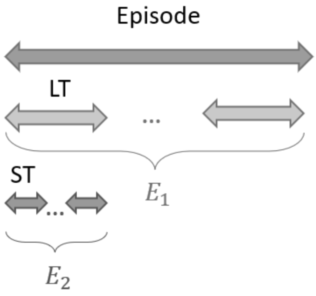


Figure 3.4. The two-phase planning model in Publication III [72].

to maximize the profit at the city scale with the overall capacity constraint. Meanwhile, the VFNs are routed to the corresponding regions with minimized traveling costs. During this process, each episode (i.e., the duration of the capacity planning horizon) is discretized into $E1$ long time (LT) slots, cf. Fig. 3.4, and we updated the global capacity plan in each LT.

During regional planning, we determined the task allocation and regional routing strategies for VFNs. More specifically, the VFNs that have already been allocated to a region are routed within the target region according to the locations of the users with minimized costs. Meanwhile, the VFNs are assigned to the users within the communication range with minimized SLA violations. During this process, each LT is discretized into $E2$ short time (ST) slots, cf. Fig. 3.4, and we updated the regional capacity plan in each ST. The two-phase capacity planning enables parallel decision-making at the regional level, which balances the execution time of the model and the techno-economic performance of the capacity plan.

Finally, the regional capacity plan was fed to the service evaluation module to measure the achieved QoS and profit. At the end of each LT, the prediction accuracy of traffic flow in the current LT is evaluated, which is used as feedback to improve the prediction accuracy in the next LT. At the end of each episode, the techno-economic performance of the successive capacity plans for that episode is evaluated, which is used as feedback for capacity planning for the next episode.

We simulated the ODVFC scenario in Helsinki for one day using the simulator described in Section 3.6. The results showed that the proposed capacity planning solution has an 8.72% higher service rate and 8.3% higher profit than the baseline solution (i.e., where the traffic flow is estimated based on Gaussian process regression [95] from the historical data). It has a 2.11% higher service rate and 79.64% higher profit than the solution solved by VRPTW [90] and a 17.38% higher service rate and 42% higher profit than the naive approach (i.e., where the VFNs randomly travel among the regions and serve the demand that is within the same region). The results also showed that the deployment of two-phase capacity planning significantly reduces the execution time compared to single-phase capacity planning (i.e., where the city is regarded as a single region), at the expense of decreased optimality in terms of profit.

3.5 MARL-based Short-term Capacity Planning

As an alternative solution to solve **RQ2**, Publication IV explores the feasibility of utilizing RL for dynamic VFN routing in the case of ODVFC. Traditional optimization approaches, such as the ILP-based ones proposed in Publications I and III, suffer from the curse of dimensionality, namely the execution time greatly increases with the size of variables. Different from the above approaches, RL can be used to solve very complex problems while addressing the curse of dimensionality [105]. Therefore, RL has the potential to reduce the time granularity of capacity planning and enable continuous VFN routing. Moreover, the proposed RL-based solution can adapt the routes of VFNs to the uncertainty in the VFC environment, such as fast-changing demand and dynamic traveling time.

We proposed to formulate the ODVFC as a MARL problem. MARL is an RL problem where multiple agents interact in a common environment. Each agent makes a decision at each time step and cooperates with other agents to achieve a common goal [86]. The goal of ODVFC is to maximize region-wide profits during an episode. In Publication IV, an episode is a fixed-length time interval, but in practice, it can also be continuous. We assumed that the VFN travels continuously through the region without entering or leaving; thus, capacity costs (i.e., rent, travel, and fuel/electric costs for all VFNs) are not affected by the VFN route. For this reason, we transformed the objective of maximizing profit into maximizing revenue.

The capacity planning, in the case of ODVFC, is converted into the dynamic routing of VFNs at the road segment level. The road network of an urban area is represented as a weighted, directed graph, where each vertex/node v is a road intersection and each edge e is a road segment [51]. We represented each road segment e as a two-dimensional vector according to the coordinates of the start and end vertices.

We formulated the dynamic routing of VFNs as an MDP with N agents, where each agent $i \in N$ is a VFN. The observation of each agent i includes two parts. One part is the set of road segments that are reachable at the next time step $Seg_{i,t+1}$, which is derived from the road network. The other part is the environmental state, which is calculated for each reachable road segment $e_2 \in Seg_{i,t+1}$. We use the environmental states at the current time step t to predict the ones at $t+1$. It includes the average traveling speed $speed(e_2)$ of all vehicles located on the road segment, the number $num(e_2)$, workload $workload(e_2)$, and latency requirements $lat(e_2)$ of all computing tasks located on the road segment, and the capacity $cap(e_2)$ of all VFNs located on the road segment. In practice, the observation of each

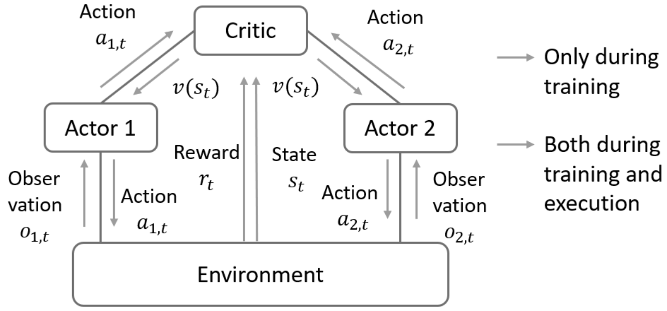


Figure 3.5. Multi-agent actor-critic framework in Publication IV, where each actor is an agent, and two actors are used for demonstration [69].

VFN is provided by the connected cellular base station. The global state is the set of observations from all agents, denoted as $s_t = \{o_{1,t}, o_{2,t}, ..., o_{N,t}\}$. The global reward r_t is the total rewards (i.e., revenue) received by all the agents during a time step t.

Previous works such as [30] set the action space as a road network, such that each VFN chooses a specific road segment as the next destination. However, this approach is not scalable for large road networks with hundreds or even thousands of road segments in real-world scenarios. As an alternative, we set the actions as the relative driving directions. More specifically, each agent can take four actions, including going straight, turning left, turning right, and making a U-turn at the end of the road. In practice, for more complex road networks with intersections consisting of more than four directions, the action set can be adjusted accordingly.

To balance the time complexity and optimality (**RQ3**), we applied the actor-critic method [66] to solve the ODVFC problem, where VFNs are agents and actors. As shown in Fig. 3.5, we used centralized training and decentralized decision-making, where each policy network (i.e., actor) generates an action distribution and a central critic network predicts discounted future returns (i.e., long-term revenue) [112] $v(s_t)$ based on the global state s_t. The central critic evaluates the cooperation among all the agents in the training phase and is not used in the execution phase (cf. Fig. 3.5). In the case of ODVFC, each VFN learns its own policy with the help of a central critic. The actor-critic approach reduces time complexity by using decentralized VFN routing strategies while enabling collaboration between VFNs by using centralized training [66].

We used the simulator presented in Section 3.6 to simulate the ODVFC scenarios during peak (i.e., 6:00 to 7:00), mid-peak (i.e., 19:00 to 20:00), and off-peak hours (i.e., 22:00 to 23:00) in the central Helsinki area, where different numbers of VFNs are deployed. The experimental results showed that the proposed solution yields 25.2% higher revenue, 25.2% more served tasks, and 17.6% lower latency than the naive approach (i.e., where the

VFNs randomly travel and serve the nearby users). The proposed solution achieves 8.3% higher revenue and 13.2% higher number of served tasks than the decentralized learning approach (i.e., where each VFN learns its own policy and critic to maximize the individual revenue). In addition, it has 40.6% and 83% lower execution time than the centralized learning method (i.e., where a global agent trains a routing policy of all VFNs to maximize the region-wide revenue) and ILP-based approach, respectively, with only 14% lower revenue than both. The results also showed that the execution time of the proposed solution increases much slower than centralized learning and ILP-based methods when the numbers of client vehicles and VFNs increase. Therefore, the proposed solution is more scalable in real-life scenarios where there is heavy traffic and large road networks.

3.6 Data-driven Simulation Platform

Due to the simplified network model in use, lack of support for fog node mobility, and limited test scenarios, existing simulation platforms [36, 100, 121] cannot provide realistic techno-economic investigations of VFC capacity plans. Therefore, we developed an open-source VFC simulator called VFogSim in Publication II, which allows real-world data as input to simulate VFC supply and demand in urban areas.

The right-hand side of Fig. 3.6 illustrates the VFogSim platform, which requires the mobility of the vehicle and the SINR map of the given area as inputs. We used Altair WinProp [39] and simulation of urban mobility (SUMO) [65] to generate synthetic data. WinProp is a network simulator that takes the 3D city database from OpenStreetMap as input and outputs the network parameters, taking into account the shadowing effects of buildings and trees. We applied the dominant path propagation model and 5G air interface configuration to estimate the SINR map for the region. SUMO is an open-source, highly portable, microscopic, and continuous traffic simulation package widely used in traffic signal control [111], autonomous driving [111], and V2X [96]. Based on the 2D road network from OpenStreetMap, we generated fine-grained microscopic traffic data in the study area. Alternatively, the inputs could be the onboard global positioning system (GPS) data and SINR data collected from real-world experiments.

We designed the VFogSim platform as a discrete-time optimizer; therefore, the input data needs to be discretized into transmission time intervals (TTIs). After inputting the data, the vehicle will be registered to different base stations (i.e., the gNBs). The network simulation module first associates vehicle trajectories with SINR values in the gNB registration block. Then, it determines the workload of different gNBs. If the client

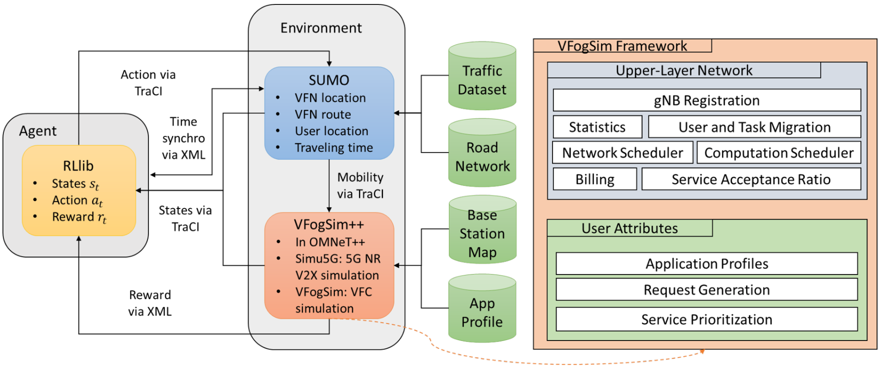


Figure 3.6. System architecture of the Ray-SUMO-VFogSim++ platform in Publication IV, which integrates RL library, Ray RLlib [79], traffic simulator, SUMO [65], 5G simulator, Simu5G [81], and VFC simulator, VFogSim++. VFogSim++ is a new version of VFogSim [3] that is embedded in OMNeT++ [107].

vehicle changes gNB, the gNB registration block triggers the user and task migration module, which processes and stores the task migration. Otherwise, the task will be moved to the network and computation schedulers. The network and computing schedulers focus on the allocation of spectrum and computing resources, respectively. During this process, we assumed that the priority of a service is determined by the requested resource, the price of the service, and the remaining service execution time. Finally, the resulting performance metrics, such as resource allocation plan, latency, and billing information, are stored separately in statistics for further techno-economic evaluation.

The VFogSim platform enables fine-grained evaluation of capacity planning solutions in urban environments. However, in VFogSim, the assessment phase only starts after the capacity planning options have been provided. To simulate the MDP formulated in Publication IV, the simulation platform is supposed to support the mobility of vehicles with 5G connectivity, the spatiotemporally varying computational capacity demands, and the deployment of different capacity planning strategies, including those based on MARL. Furthermore, different modules need to interact with each other and work in an online manner.

Since none of the existing simulators [111, 96] supports all of the above features, we built an extended version of VFogSim, referred to as Ray-SUMO-VFogSim++, in Publication IV. It improves upon existing simulators and accommodates four open-source platforms. Specifically, we used SUMO [65] for vehicular mobility simulation. Simu5G [81] is used for 5G network simulation, which is the evolution of the popular SimuLTE [108] that incorporates 5G NR access. We used VFogSim++ for VFC simulations, which is a new version of VFogSim [3] that is embedded in OMNeT++ [107] and located in the same package as Simu5G. Finally, Ray RLlib [79], a widely-used RL library, is used for building the MARL environment.

Fig. 3.6 shows our proposed simulation platform Ray-SUMO-VFogSim++. In this platform, the agents (i.e., VFNs) are modeled in Ray RLlib. The environment is simulated in SUMO and VFogSim++, which take charge of vehicular mobility and network simulations, respectively. The agents interact with SUMO through traffic control interface (TraCI) [24]. They interact with VFogSim++ through extensible markup language (XML), where one program writes and the other reads at each time. During our simulation, the vehicular simulation in Ray RLlib is updated per simulation step t (e.g., 1 second) while the network simulation in VFogSim++ is updated per TTI (e.g., 0.1 second), thus a timer is used for time synchronization.

The VFogSim and Ray-SUMO-VFogSim++ [76] are made open-source for future research in VFC. They both follow a modular and customizable design and support various application scenarios. They can be used to evaluate the techno-economic performance of different deployment scenarios. They can also be used to analyze the effectiveness of diverse network and computation scheduling algorithms, pricing strategies, and prioritization mechanisms. In particular, the 5G NR V2X module in Ray-SUMO-VFogSim++ provides a realistic simulation of wireless vehicular communication and task offloading. Furthermore, the embedded MARL environment supports the evaluation of various RL-based algorithms, including task allocation, capacity planning, and vehicle routing.

4. Open Questions

This chapter discusses potential areas for future research based on the results of this dissertation.

4.1 Resource Heterogeneity

Compared to cloud computing where there is sufficient capacity, it is challenging to understand the performance limitations and bottlenecks of vehicular applications running on fog nodes, due to their heterogeneity and constrained capacity [93]. Thus, to obtain realistic application profiles, a benchmark suite is necessary for the VFC environment, taking into account resource heterogeneity. Besides, edge federation refers to an integrated resource provisioning model, which seamlessly realizes resource cooperation and service provisioning across standalone edge computing providers and clouds [13]. Thus, edge federation can be utilized for resource allocation and capacity planning for VFC, to efficiently schedule and utilize the resources over CFNs, VFNs, and the cloud.

4.2 Task Heterogeneity

This dissertation does not consider the variations in task arrival rate (i.e., how many requests would be generated by each vehicle) [70] or user preference (which services or applications each vehicle would prefer to use) [46]. We assume that they both follow a uniform distribution. In future work, we will take the uncertainty in these aspects into account. For example, queuing theory can be combined with the demand prediction model in capacity planning [84]. A queuing system consists of the arrival process, server, queue, service discipline, service time distribution, departure process, and system performance measures. Therefore, the queue models can be used to include different task arrival rate distributions and vehicular application combinations.

4.3 Caching for VFC

In this dissertation, only computation offloading is considered in terms of capacity planning, whereas caching can also be studied for VFC [106]. In VFC, the client vehicle can be both a data provider and a data receiver. If we cache popular data on CFNs and VFNs that are closer to the client vehicle, this would save time for the client vehicles to access the data and reduce the congestion in the cloud [64]. Therefore, an efficient caching strategy is needed that can intelligently recognize similar processing tasks and cache them for future requests. Furthermore, how to jointly decide on cache placement, VFN scheduling or routing strategies, and computing resource allocation will be a challenge.

4.4 Carriers of VFNs

This dissertation considers vehicles such as buses and taxis to serve as VFNs. For taxis, we assume that they have flexible routes when there is no customer. Whereas, in real life, the availability of taxis needs to be considered, especially during large events and peak hours. In addition, the locations of charging or refueling stations need to be considered when planning the routes for the VFNs [33]. Another potential approach is to deploy VFNs on wireless charging electric buses, which receive power from underground transmitters during moving [34].

Alternatively, the UAVs can be served as VFNs. The UAVs can freely move in continuous space instead of being restricted by road networks, or they can be designated for the VFC service, which addresses the availability issue. Furthermore, they will not increase vehicular traffic congestion during peak hours. However, due to the size and weight constraints, the computing capacity of UAVs is limited [67]. Another limitation of UAVs is the high energy consumption and short battery life [103, 67]. Furthermore, there is a stringent time requirement for the UAV control [103, 117].

4.5 Optimization Objective

This dissertation considers a trade-off between the techno-economic performance of the capacity plan and the time complexity of the capacity planning model, and the QoS has been represented as latency. However, we did not consider the different QoS requirements of artificial intelligence (AI) based services. In addition to latency, one AI-based service may use different neural networks corresponding to different accuracy levels and computational complexity. In the future, the QoS can be extended to cover both latency and accuracy [59].

In addition, if we use machine learning for spatio-temporal feature extraction and demand prediction, the key challenges include the support for heterogeneous spatio-temporal data as well as efficient and scalable computation [61]. Furthermore, if we deploy deep RL for capacity planning, another research direction is how to make it more resource efficient [48].

4.6 Security and Privacy

If we use VFC for data sharing, the users' sensitive information, such as identity and location, as well as social graphs are entirely exposed to the service provider, thereby causing critical security and privacy threats [58]. From a security perspective, confidentiality, integrity, authentication, and location validations are essential when planning for VFC [40]. From a privacy perspective, the privacy of the vehicle's information, such as personal and location information, needs to be preserved during training and data sharing [40].

Federated learning (FL) refers to a learning strategy that decouples model training from direct access to the raw data. In FL, the learning task is solved by a loose federation of participating devices coordinated by a central server [49]. The participating devices do not share the raw data with the central server, but an update of the trained model instead, to decrease communication costs and preserve individual privacy. If we use FL for training the models in VFC, one of the major concerns is how to minimize the rounds of communication during training, while balancing accuracy at the same time [60].

5. Conclusion

In recent years, autonomous driving has attracted great attention from both academia and industry. To meet the computational demand generated by compute-intensive and latency-critical vehicular applications, fog computing reduces the latency by moving the computational resources close to where data is generated. VFC is an emerging computing paradigm where fog nodes deployed on moving vehicles complement stationary fog nodes to satisfy the spatiotemporally varying demand for computing resources in a cost-efficient manner. However, due to the high mobility of vehicles and the uncertainty in the vehicular traffic environment, some challenges remain to be addressed before the real-world implementation of VFC. In particular, we focus on capacity planning for VFC, taking into account the dynamic demand and supply of computational resources.

To address the research questions, this dissertation has followed a data-driven methodology to develop capacity planning frameworks for VFC. First, to address spatio-temporal dynamics in vehicular traffic, we propose an ILP-based long-term capacity planning framework that determines the location and number of CFNs together with the schedules and routes of VFNs carried by buses. Second, to address the uncertainty in vehicular traffic, we present an ILP-based on-demand capacity planning framework based on traffic prediction, which decides the number and routes of VFNs carried by taxis through city-scale and regional planning. Furthermore, we propose a MARL-based on-demand capacity planning framework, which focuses on dynamically routing the VFNs carried by taxis using the actor-critic method. Finally, we develop an open-source VFC simulation platform, which takes real-world vehicular and network data as inputs and simulates the supply and demand of computational resources in urban environments.

The evaluation of the developed frameworks indicates their applicability for enabling a QoS-guaranteed and cost-efficient VFC service. We believe that the results of this work can contribute to the development of VFC for next-generation vehicular applications in the future.

References

[1] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification. Technical Specification (TS) 36.331, 3rd Generation Partnership Project (3GPP), April 2017. Version 14.2.2.

[2] 3GPP. Study on enhancement of 3GPP support for 5G V2X services. Technical Report 22.886, 3rd Generation Partnership Project (3GPP), 2018. Version 15.

[3] Özgür Umut Akgül, Wencan Mao, Byungjin Cho, and Yu Xiao. VFogSim: A data-driven platform for simulating vehicular fog computing environment. *IEEE Systems Journal*, 17(3):5002–5013, 2023.

[4] Andreas Andreou, Constandinos X. Mavromoustakis, Jordi Mongay Batalla, Evangelos K. Markakis, George Mastorakis, and Shahid Mumtaz. UAV trajectory optimisation in smart cities using modified A* algorithm combined with haversine and vincenty formulas. *IEEE Transactions on Vehicular Technology*, pages 1–13, 2023.

[5] Hamidreza Bagheri, Md Noor-A-Rahim, Zilong Liu, Haeyoung Lee, Dirk Pesch, Klaus Moessner, and Pei Xiao. 5G NR-V2X: Toward connected and cooperative autonomous driving. *IEEE Communications Standards Magazine*, 5(1):48–54, 2021.

[6] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, page 13–16, 2012.

[7] Abdelwahab Boualouache, Ridha Soua, and Thomas Engel. Toward an SDN-based data collection scheme for vehicular fog computing. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020.

[8] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[9] Antonio Brogi and Stefano Forti. QoS-aware deployment of IoT applications through the fog. *IEEE Internet of Things Journal*, 4(5):1185–1192, 2017.

[10] Antonio Brogi, Stefano Forti, and Ahmad Ibrahim. How to best deploy your fog applications, probably. In *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pages 105–114, 2017.

[11] Toon Calders and Bart Custers. *What Is Data Mining and How Does It Work?*, pages 27–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[12] Rodrigo Calheiros, R. Ranjan, Cesar De Rose, and Rajkumar Buyya. CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *Computing Research Repository*, 04 2009.

[13] Xiaofeng Cao, Guoming Tang, Deke Guo, Yan Li, and Weiming Zhang. Edge federation: Towards an integrated service provisioning model. *CoRR*, abs/1902.09055, 2019.

[14] Luca Maria Castiglione, Paolo Falcone, Alberto Petrillo, Simon Pietro Romano, and Stefania Santini. Cooperative intersection crossing over 5G. *IEEE/ACM Transactions on Networking*, 29(1):303–317, 2021.

[15] Xiaosha Chen, Supeng Leng, Ke Zhang, and Kai Xiong. A machine-learning based time constrained resource allocation scheme for vehicular fog computing. *China Communications*, 16(11):29–41, 2019.

[16] Te-Chuan Chiu, Wei-Ho Chung, Ai-Chun Pang, Ya-Ju Yu, and Pei-Hsuan Yen. Ultra-low latency service provision in 5G fog-radio access networks. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, 2016.

[17] Byungjin Cho and Yu Xiao. Learning-based decentralized offloading decision making in an adversarial environment. *IEEE Transactions on Vehicular Technology*, 70(11):11308–11323, 2021.

[18] CellMapper contributors. CellMapper, 2023. Last accessed 1 February 2023, https://www.cellmapper.net.

[19] Fintraffic contributors. Traffic measurement system data, 2023. Last accessed 1 February 2023, https://www.digitraffic.fi/en/road-traffic/lam/.

[20] HERE WeGo contributors. HERE Traffic API, 2023. Last accessed 1 February 2023, https://developer.here.com/documentation/traffic.

[21] HERE WeGo contributors. HERE WeGo, 2023. Last accessed 1 February 2023, https://wego.here.com.

[22] OpenCellid contributors. OpenCellid, 2023. Last accessed 1 February 2023, https://opencellid.org/.

[23] OpenStreetMap contributors. OpenStreetMap, 2023. Last accessed 1 February 2023, https://www.openstreetmap.org.

[24] SUMO contrubutors. Traffic control interface (TraCI), 2023. Last accessed 1 February 2023, https://sumo.dlr.de/docs/TraCI.html.

[25] HSL corporation. Helsinki region transport open data, 2023. Last accessed 1 February 2023, https://www.hsl.fi/en/hsl/open-data.

[26] Antonio Coutinho, Fabiola Greve, Cassio Prazeres, and Joao Cardoso. Fogbed: A rapid-prototyping emulation environment for fog computing. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, 2018.

[27] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2020.

[28] Penglin Dai, Kaiwen Hu, Xiao Wu, Huanlai Xing, and Zhaofei Yu. Asynchronous deep reinforcement learning for data-driven task offloading in MEC-empowered vehicular networks. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[29] Xiaolei Di, Yu Xiao, Chao Zhu, Yang Deng, Qinpei Zhao, and Weixiong Rao. Traffic congestion prediction by spatiotemporal propagation patterns. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pages 298–303, 2019.

[30] Rong Ding, Zhaoxing Yang, Yifei Wei, Haiming Jin, and Xinbing Wang. Multi-agent reinforcement learning for urban crowd sensing with for-hire vehicles. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[31] Hao Du, Supeng Leng, Fan Wu, Xiaosha Chen, and Sun Mao. A new vehicular fog computing architecture for cooperative sensing of autonomous driving. *IEEE Access*, 8:10997–11006, 2020.

[32] Lily Elefteriadou. *An Introduction to Traffic Flow Theory*, volume 84. Springer Optimization and Its Applications, Jan. 2014.

[33] Guiyun Fan, Haiming Jin, Yiran Zhao, Yiwen Song, Xiaoying Gan, Jiaxin Ding, Lu Su, and Xinbing Wang. Joint order dispatch and charging for electric self-driving taxi systems. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pages 1619–1628, 2022.

[34] Arman Fathollahi, Meysam Gheisarnejad, Jalil Boudjadar, Maryam Homayounzadeh, and Mohammad-Hassan Khooban. Optimal design of wireless charging electric buses-based machine learning: A case study of nguyen-dupuis network. *IEEE Transactions on Vehicular Technology*, pages 1–9, 2023.

[35] Xiaohu Ge, Hui Cheng, Guoqiang Mao, Yang Yang, and Song Tu. Vehicular communications for 5G cooperative small-cell networks. *IEEE Transactions on Vehicular Technology*, 65(10):7882–7894, 2016.

[36] Harshit Gupta, Amir Dastjerdi, Soumya Ghosh, and Rajkumar Buyya. iFogSim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments. *Software: Practice and Experience*, 47, 06 2016.

[37] Faisal Haider, Decheng Zhang, Marc St-Hilaire, and Christian Makaya. On the planning and design problem of fog computing networks. *IEEE Transactions on Cloud Computing*, 9(2):724–736, 2021.

[38] Najmul Hassan, Kok-Lim Alvin Yau, and Celimuge Wu. Edge computing in 5G: A review. *IEEE Access*, 7:127276–127289, 2019.

[39] Reiner Hoppe, Gerd Wölfle, and Ulrich Jakobus. Wave propagation and radio network planning software winprop added to the electromagnetic solver package feko. In *2017 International Applied Computational Electromagnetics Society Symposium - Italy (ACES)*, pages 1–2, 2017.

[40] Mohammad Aminul Hoque and Ragib Hasan. Towards an analysis of the architecture, security, and privacy issues in vehicular fog computing. In *2019 SoutheastCon*, pages 1–8, 2019.

[41] Xiangwang Hou, Zhiyuan Ren, Jingjing Wang, Wenchi Cheng, Yong Ren, Kwang-Cheng Chen, and Hailin Zhang. Reliable computation offloading for edge-computing-enabled software-defined IoV. *IEEE Internet of Things Journal*, 7(8):7097–7111, 2020.

[42] Xueshi Hou, Yong Li, Min Chen, Di Wu, Depeng Jin, and Sheng Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, 2016.

[43] Han Hu, Zuan Chen, Fuhui Zhou, Zhu Han, and Hongbo Zhu. Joint resource and trajectory optimization for heterogeneous-UAVs enabled aerial-ground cooperative computing networks. *IEEE Transactions on Vehicular Technology*, pages 1–15, 2023.

[44] Cheng Huang, Rongxing Lu, and Kim-Kwang Raymond Choo. Vehicular fog computing: Architecture, use case, and security and forensic challenges. *IEEE Communications Magazine*, 55(11):105–111, 2017.

[45] Haichao Huang, Jingya Chen, Rui Sun, and Shuang Wang. Short-term traffic prediction based on time series decomposition. *Physica A: Statistical Mechanics and its Applications*, 585:126441, 09 2021.

[46] Xiaoge Huang, Zhi Chen, Qianbin Chen, and Jie Zhang. Federated learning based QoS-aware caching decisions in fog-enabled Internet of Things networks. *Digital Communications and Networks*, 2022.

[47] Md. Muzakkir Hussain, Mohammad Saad Alam, and M.M. Sufyan Beg. Vehicular fog computing-planning and design. In *International Conference on Computational Intelligence and Data Science*, volume 167, pages 2570–2580, 2020.

[48] Ingook Jang, Hyunseok Kim, Donghun Lee, Young-Sung Son, and Seonghyun Kim. Knowledge transfer for on-device deep reinforcement learning in resource constrained edge computing systems. *IEEE Access*, 8:146588–146597, 2020.

[49] Xiaopeng Jiang, Thinh On, NhatHai Phan, Hessamaldin Mohammadi, Vijaya Datta Mayyuri, An Chen, Ruoming Jin, and Cristian Borcea. Zone-based federated learning for mobile sensing data. In *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2023. ISSN: 2474-249X.

[50] Joelle Klaimi, Sidi-Mohammed Senouci, and Mohamed-Ayoub Messous. Theoretical game approach for mobile users resource management in a vehicular fog computing environment. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 452–457, 2018.

[51] Ton Kloks and Mingyu Xiao. *A Guide to Graph Algorithms*. Springer Verlag, Singapore, Mar. 2022.

[52] Yousri Kouki and Thomas Ledoux. SLA-driven capacity planning for cloud applications. In *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pages 135–140, 2012.

[53] S. Vasantha Kumar and Lelitha Devi Vanajakshi. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review*, 7, 2015.

[54] Giuseppe Lancia and Paolo Serafini. *Integer Linear Programming*, pages 43–66. 01 2018.

[55] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop: Rapid prototyping for software-defined networks. In *Hotnets-IX: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, number 19, pages 1–6, 2010.

[56] Seung-Seob Lee and Sukyoung Lee. Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information. *IEEE Internet of Things Journal*, 7(10):10450–10464, 2020.

[57] Jingwen Li, Yining Ma, Ruize Gao, Zhiguang Cao, Andrew Lim, Wen Song, and Jie Zhang. Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. *IEEE Transactions on Cybernetics*, 52(12):13572–13585, 2022.

[58] Meng Li, Liehuang Zhu, Zijian Zhang, Xiaojiang Du, and Mohsen Guizani. PROS: A privacy-preserving route-sharing service via vehicular fog computing. *IEEE Access*, 6:66188–66197, 2018.

[59] Xuebing Li, Byungjin Cho, and Yu Xiao. Balancing latency and accuracy on deep video analytics at the edge. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 299–306, 2022.

[60] Yiran Li, Hongwei Li, Guowen Xu, Tao Xiang, and Rongxing Lu. Practical privacy-preserving federated learning in vehicular fog computing. *IEEE Transactions on Vehicular Technology*, 71(5):4692–4705, 2022.

[61] Kaiqi Liu, Panrong Tong, Mo Li, Yue Wu, and Jianqiang Huang. ST4ML: Machine learning oriented spatio-temporal data processing at scale. In *ACM SIGMOD/PODS 2023*, pages 1–28, 2023.

[62] Shuncheng Liu, Han Su, Yan Zhao, Kai Zeng, and Kai Zheng. Lane change scheduling for autonomous vehicle: A prediction-and-search framework. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 3343–3353, 2021.

[63] Yi Liu, Huimin Yu, Shengli Xie, and Yan Zhang. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. *IEEE Transactions on Vehicular Technology*, 68(11):11158–11168, 2019.

[64] Ying Liu, Qiang He, Dequan Zheng, Mingwei Zhang, Feifei Chen, and Bin Zhang. Data caching optimization in the edge computing environment. In *2019 IEEE International Conference on Web Services (ICWS)*, pages 99–106, 2019.

[65] Pablo Alvarez Lopez, Michael Behrisch, and Laura Bieker-Walz. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, 2018.

[66] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6382–6393, 2017.

[67] Quyuan Luo, Tom H. Luan, Weisong Shi, and Pingzhi Fan. Deep reinforcement learning based computation offloading and trajectory planning for multi-UAV cooperative target search. *IEEE Journal on Selected Areas in Communications*, pages 1–1, 2022.

[68] Xianglong Luo, Danyang Li, Yu Yang, and Shengrui Zhang. Spatiotemporal traffic flow prediction with KNN and LSTM. *Journal of Advanced Transportation*, 2019:1–10, 02 2019.

[69] Tianle Mai, Haipeng Yao, Zehui Xiong, Song Guo, and Dusit Tao Niyato. Multi-agent actor-critic reinforcement learning based in-network load balance. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.

[70] Sanaullah Manzoor, Adnan Noor Mian, Ahmed Zoha, and Muhammad Ali Imran. Federated learning empowered mobility-aware proactive content offloading framework for fog radio access networks. *Future Generation Computer Systems*, 133:307–319, 2022.

[71] Wencan Mao. Phd forum abstract: Capacity planning for vehicular fog computing. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 186–187, 2022.

[72] Wencan Mao, Ozgur Umut Akgul, Byungjin Cho, Yu Xiao, and Antti Ylä-Jääski. On-demand vehicular fog computing for beyond 5G networks. *IEEE Transactions on Vehicular Technology*, pages 1–17, 2023.

[73] Wencan Mao, Ozgur Umut Akgul, Abbas Mehrabi, Byungjin Cho, Yu Xiao, and Antti Ylä-Jääski. Data-driven capacity planning for vehicular fog computing. *IEEE Internet of Things Journal*, 9(15):13179–13194, 2022.

[74] Markets and Markets. Self-driving cars market by component, vehicle, level of autonomy, mobility type, EV and region - global forecast to 2030, 2022. Last accessed 1 February 2023, https://www.researchandmarkets.com/reports/4240726/self-driving-cars-market-by-component-radar.

[75] Ruben Mayer, Leon Graser, Harshit Gupta, Enrique Saurez, and Umakishore Ramachandran. EmuFog: Extensible and scalable emulation of large-scale fog computing infrastructures. In *2017 IEEE Fog World Congress (FWC)*, pages 1–6, 2017.

[76] Aalto mc2: Mobile Cloud Computing. DataFog: A data-driven platform for capacity and resource management in vehicular fog computation, 2023. Last accessed 1 February 2023, https://mobilecloud.aalto.fi/?page_id=1441.

[77] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

[78] Rafael Molina-Masegosa and Javier Gozalvez. LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications. *IEEE Vehicular Technology Magazine*, 12(4):30–39, 2017.

[79] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging AI applications. In *OSDI'18: 13th USENIX Symposium on Operating Systems Design and Implementation*, page 561–577, 2018.

[80] Gaurang Naik, Biplav Choudhury, and Jung-Min Park. IEEE 802.11bd & 5G NR V2X: Evolution of radio access technologies for V2X communications. *IEEE Access*, 7:70169–70184, 2019.

[81] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Virdis. Simu5G–an OMNeT++ library for end-to-end performance evaluation of 5G networks. *IEEE Access*, 8:181176–181191, 2020.

[82] MohammadReza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takac. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[83] Zhaolong Ning, Jun Huang, and Xiaojie Wang. Vehicular fog computing: Enabling real-time traffic management for smart cities. *IEEE Wireless Communications*, 26(1):87–93, 2019.

[84] Marius Noreikis, Yu Xiao, and Yuming Jiang. Edge capacity planning for real time compute-intensive applications. In *2019 IEEE International Conference on Fog Computing (ICFC)*, pages 175–184, 2019.

[85] Marius Noreikis, Yu Xiao, and Antti Ylä-Jaäiski. QoS-oriented capacity planning for edge computing. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017.

[86] Afshin Oroojlooyjadid and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *CoRR*, abs/1908.03963, 2019.

[87] Joaquín Ortega, Nelva Almanza-Ortega, Andrea Vega-Villalobos, Rodolfo Pazos-Rangel, José Crispin Zavala-Diaz, and Alicia Martínez-Rebollar. *The K-Means Algorithm Evolution*. 04 2019.

[88] Bo Peng, Jiahai Wang, and Zizhen Zhang. A deep reinforcement learning algorithm using dynamic attention model for vehicle routing problems. *CoRR*, abs/2002.03282, 2020.

[89] Xiting Peng, Kaoru Ota, and Mianxiong Dong. Multiattribute-based double auction toward resource allocation in vehicular fog computing. *IEEE Internet of Things Journal*, 7(4):3094–3103, 2020.

[90] Krupa Prag, Matthew Woolway, and Byron A. Jacobs. Optimising the vehicle routing problem with time windows under standardised metrics. In *2019 6th International Conference on Soft Computing and Machine Intelligence (ISCMI)*, pages 111–115, 2019.

[91] Gopika Premsankar, Bissan Ghaddar, Mario Di Francesco, and Rudi Verago. Efficient placement of edge computing devices for vehicular applications in smart cities. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2018.

[92] Tariq Qayyum, Asad Waqar Malik, Muazzam A. Khan Khattak, Osman Khalid, and Samee U. Khan. FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access*, 6:63570–63583, 2018.

[93] Kaustubh Rajendra Rajput, Chinmay Dilip Kulkarni, Byungjin Cho, Wei Wang, and In Kee Kim. EdgeFaaSBench: Benchmarking edge devices using serverless computing. In *2022 IEEE International Conference on Edge Computing and Communications (EDGE)*, pages 93–103, 2022.

[94] Nadia Ranaldo and Eugenio Zimeo. Capacity–driven utility model for service level agreement negotiation of cloud services. *Future Generation Computer Systems*, 55, 04 2015.

[95] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005.

[96] Max Schettler, Dominik S. Buse, Anatolij Zubow, and Falko Dressler. How to train your ITS? integrating machine learning with vehicular network simulation. In *2020 IEEE Vehicular Networking Conference (VNC)*, pages 1–4, 2020.

[97] Gaoqing Shen, Lei Lei, Xinting Zhang, Zhilin Li, Shengsuo Cai, and Lijuan Zhang. Multi-UAV cooperative search based on reinforcement learning with a digital twin driven training framework. *IEEE Transactions on Vehicular Technology*, pages 1–15, 2023.

[98] Jagdeep Singh, Parminder Singh, Mustapha Hedabou, and Neeraj Kumar. An efficient machine learning-based resource allocation scheme for SDN-enabled fog computing environment. *IEEE Transactions on Vehicular Technology*, pages 1–15, 2023.

[99] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, 2011.

[100] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. EdgeCloudSim: An environment for performance evaluation of edge computing systems. In *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 39–44, 2017.

[101] Ivan Stojmenovic, Sheng Wen, Xinyi Huang, and Hao Luan. An overview of fog computing and its security issues. *Concurrency and Computation: Practice and Experience*, 28(10):2991–3005, 2016.

[102] Ioanna Stypsanelli, Olivier Brun, Samir Medjiah, and Balakrishna J. Prabhu. Capacity planning of fog computing infrastructures under probabilistic delay guarantees. In *2019 IEEE International Conference on Fog Computing (ICFC)*, pages 185–194, 2019.

[103] Hui Sun, Bo Zhang, Xiuye Zhang, Ying Yu, Kewei Sha, and Weisong Shi. FlexEdge: Dynamic task scheduling for a UAV-based on-demand mobile edge server. *IEEE Internet of Things Journal*, 9(17):15983–16005, 2022.

[104] Ali Sunyaev. *Fog and Edge Computing*, pages 237–264. Springer International Publishing, Cham, 2020.

[105] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[106] Le Thanh Tan and Rose Qingyang Hu. Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 67(11):10190–10203, 2018.

[107] András Varga and Rudolf Hornig. An overview of the OMNeT++ simulation environment. In *1st international conference on Simulation tools and techniques for communications*, page 60, 01 2008.

[108] Antonio Virdis, Giovanni Stea, and Giovanni Nardini. SimuLTE - a modular system-level simulator for lte/lte-a networks based on omnet++. In *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*, pages 59–70, 2014.

[109] Zhe Wang, Zhangdui Zhong, and Minming Ni. Application-aware offloading policy using SMDP in vehicular fog computing systems. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2018.

[110] Philip Wette, M Draxler, Arne Schwabe, F Wallaschek, M Zahraee, and H Karl. MaxiNet: Distributed emulation of software-defined networks. In *2014 IFIP Networking Conference, IFIP Networking 2014*, pages 1–9, 06 2014.

[111] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M. Bayen. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *CoRR*, abs/1710.05465, 2017.

[112] Yu Xiao and Chao Zhu. Vehicular fog computing: Vision and challenges. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 6–9, 2017.

[113] Jindou Xie, Yunjian Jia, Zhengchuan Chen, Zhaojun Nan, and Liang Liang. Efficient task completion for parallel offloading in vehicular fog computing. *China Communications*, 16(11):42–55, 2019.

[114] Qinge Xie, Tiancheng Guo, Yang Chen, Yu Xiao, Xin Wang, and Ben Y. Zhao. "how do urban incidents affect traffic speed?" a deep graph convolutional network for incident-driven traffic speed prediction. *ArXiv*, abs/1912.01242, 2019.

[115] Chen Xu, Yahui Wang, Zhenyu Zhou, Bo Gu, Valerio Frascolla, and Shahid Mumtaz. A low-latency and massive-connectivity vehicular fog computing framework for 5G. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2018.

[116] Rahul Yadav, Weizhe Zhang, Omprakash Kaiwartya, Houbing Song, and Shui Yu. Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing. *IEEE Transactions on Vehicular Technology*, 69(12):14198–14211, 2020.

[117] Jun Yang, Xinghui You, Gaoxiang Wu, Mohammad Mehedi Hassan, Ahmad Almogren, and Joze Guna. Application of reinforcement learning in UAV cluster task scheduling. *Future Generation Computer Systems*, 95(C):140–148, jun 2019.

[118] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: Concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*, Mobidata '15, page 37–42, 2015.

[119] Jiaming Yin, Weixiong Rao, and Chenxi Zhang. Learning shortest paths on large dynamic graphs. In *2021 22nd IEEE International Conference on Mobile Data Management (MDM)*, pages 201–208, 2021.

[120] Quan Yuan, Jinglin Li, Haibo Zhou, Tao Lin, Guiyang Luo, and Xuemin Shen. A joint service migration and mobility optimization approach for vehicular edge computing. *IEEE Transactions on Vehicular Technology*, 69(8):9041–9052, 2020.

[121] Xuezhi Zeng, Saurabh Kumar Garg, Peter Strazdins, Prem Prakash Jayaraman, Dimitrios Georgakopoulos, and Rajiv Ranjan. IOTSim: A simulator for analysing iot applications. *Journal of Systems Architecture*, 72:93–107, 2017.

[122] Decheng Zhang, Faisal Haider, Marc St-Hilaire, and Christian Makaya. Model and algorithms for the planning of fog computing networks. *IEEE Internet of Things Journal*, 6(2):3873–3884, 2019.

[123] Ke Zhang, Fang He, Zhengchao Zhang, Xi Lin, and Meng Li. Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 121:102861, 2020.

[124] Junhui Zhao, Ming Kong, Qiuping Li, and Xiaoke Sun. Contract-based computing resource management via deep reinforcement learning in vehicular fog computing. *IEEE Access*, 8:3319–3329, 2020.

[125] Zibin Zheng, Yatao Yang, Jiahao Liu, Hong-Ning Dai, and Yan Zhang. Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3927–3939, 2019.

[126] Zhenyu Zhou, Haijun Liao, Xiaoyan Wang, Shahid Mumtaz, and Jonathan Rodriguez. When vehicular fog computing meets autonomous driving: Computational resource management and task offloading. *IEEE Network*, 34(6):70–76, 2020.

[127] Zhenyu Zhou, Haijun Liao, Xiongwen Zhao, Bo Ai, and Mohsen Guizani. Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty. *IEEE Transactions on Vehicular Technology*, 68(9):8322–8335, 2019.

[128] Zhenyu Zhou, Pengju Liu, Junhao Feng, Yan Zhang, Shahid Mumtaz, and Jonathan Rodriguez. Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach. *IEEE Transactions on Vehicular Technology*, 68(4):3113–3125, 2019.

[129] Chao Zhu, Yi-Han Chiang, Abbas Mehrabi, Yu Xiao, Antti Ylä-Jääski, and Yusheng Ji. Chameleon: Latency and resolution aware task offloading for visual-based assisted driving. *IEEE Transactions on Vehicular Technology*, 68(9):9038–9048, 2019.

[130] Chao Zhu, Yi-Han Chiang, Yu Xiao, and Yusheng Ji. Flexsensing: A QoI and latency-aware task allocation scheme for vehicle-based visual crowdsourcing via deep Q-network. *IEEE Internet of Things Journal*, 8(9):7625–7637, 2021.

[131] Chao Zhu, Giancarlo Pastor, Yu Xiao, and Antti Ylajaaski. Vehicular fog computing for video crowdsourcing: Applications, feasibility, and challenges. *IEEE Communications Magazine*, 56(10):58–63, 2018.

[132] Chao Zhu, Jin Tao, Giancarlo Pastor, Yu Xiao, Yusheng Ji, Quan Zhou, Yong Li, and Antti Ylä-Jääski. Folo: Latency and quality optimized task allocation in vehicular fog computing. *IEEE Internet of Things Journal*, 6(3):4150–4161, 2019.

Errata

Publication I

The notations "$\in (0,1)$" should be "$\in \{0,1\}$" in Equations (1e), (2d), and (3d).

Aalto-DT 172/2023

This dissertation focuses on capacity planning for VFC. The objective of capacity planning is to maximize the techno-economic performance of VFC in terms of profit and QoS. To address the spatial-temporal dynamics of vehicular traffic, this dissertation presents a capacity planning solution for VFC that jointly decide the location and number of CFNs together with the route and schedule of VFNs carried by buses. Such a long-term planning solution is supposed to be updated seasonally according to the traffic pattern and bus timetables. To address the uncertainty in the computational resource demand, this dissertation presents two capacity planning solutions for VFC that dynamically schedule the routes of VFNs carried by taxis in an on-demand manner. Such a short-term planning solution is supposed to be updated within minutes or even seconds.

To evaluate the techno-economic performance of our capacity planning solutions, an open-source simulator was developed that takes real-world data as inputs and simulates the VFC scenarios in urban environments. The results of this dissertation can contribute to the development of edge and fog computing, the Internet of Vehicles (IoV), and intelligent transportation systems (ITS).

9 789526 414812

ISBN 978-952-64-1481-2 (printed)
ISBN 978-952-64-1482-9 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Electrical Engineering
Department of Information and Communications Engineering
www.aalto.fi